

11

الهاكر الأخلاقي

Session Hijacking

By

Dr.Mohammed Sobhy Teba

Session Hijacking

https://www.facebook.com/tibea2004

CONTENTS

11.1 بعض المفاهيم الأساسية (Session Hijacking Concept)

من أجل فهم **session hijacking** وكيفية استخدام هذه الطريقة من اجل القرصنة، لذا يجب عليك أن تكون على دراية بالمفاهيم الأساسية لـ **session hijacking**.

هذا القسم يسلط الضوء على **session hijacking** والمخاطر الناجمة عنها، والتقنيات المستخدمة في **session hijacking**، **spoofing** مقابل **hijacking**، عملية **session hijacking**، وأنواع **session hijacking**، و**session hijacking** في نموذج **OSI**.

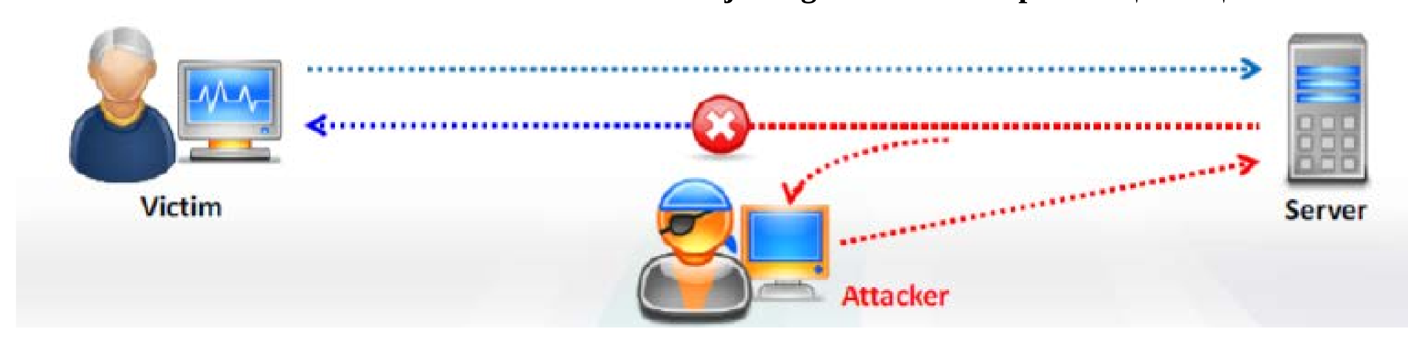
ما هو session hijacking؟

أحيانا نقوم بالدخول الى بعض المواقع ونقوم بإدخال كلمات السر إذا كان الموقع يتطلب هذا او حتى مجرد الدخول الى صفحات داينمك تتولد جلسة او ما يسمى **Session** وفائدة هذه الجلسة عند التنقل من لصفحة الى صفحة اخرى تجعلك لا تفقد معلوماتك كزائر او عضو او أيا كان حسب اعدادات المبرمج. مثال اخر، عند تسجل دخولك في المنتدى تتنقل بين المواضيع ولكنك لا تضطر الى اعادة تسجيل دخولك في صفحات المنتدى الاخرى لأنه تم انشاء جلسة لك بمجرد اغلاق المتصفح تتدمر الجلسة.

Session Hijacking أو ما يعرف باسم اقتحام الجلسات يشير الى الاستيلاء على الجلسة لاستغلال جلسة كمبيوتر صالحة حيث يأخذ أحد المهاجمين عبر جلسة عمل بين جهازي كمبيوتر. المهاجم يسرق الجلسة الصالحة والتي يتم استخدامها للوصول الى النظام واستخراج البيانات وطبعاً لها انواع حسب طريقة المهاجم. **TCP session hijacking** يعني السيطرة على جلسة **TCP** المتبادلة بين جهازي كمبيوتر. ويتم ذلك من خلال حزم **source-routed IP packets**. يمكن للمهاجم الذي تم تسجيل دخوله إلى النظام ان يشارك في المحادثة مع المستخدمين الآخرين على الأنظمة الأخرى من خلال تحويل الحزم إلى النظام الخاص به. الاستيلاء الأعمى "**Blind hijacking**" هو طريقة أخرى والتي من خلالها يكون الردود على النظام يمكن ان يفترض. هجوم رجل في الوسط (**MITM**) هو طريقة أخرى للذي يستخدم **sniffing** لتعقب محادثة بين اثنين من المستخدمين. يتم تنفيذ هجمات الحرمان من الخدمة **DDoS** بحيث يتعطل النظام، الأمر الذي يؤدي إلى خسارة كبيره في الحزم.

خطوات الاستيلاء على الجلسة "Steps in session hijacking":

1- تتبع الاتصال "**Tracking the connection**".
2- إعادة مزامنة الاتصال "**Desynchronizing the connection**".
3- حقن حزم المهاجم "**Injecting the attacker's packet**".



المخاطر التي يشكلها Hijacking

Hijacking بسيط جدا لإطلاقه. معظم أجهزة الكمبيوتر تعتبر نقطة ضعف اما **Hijacking** وذلك لأنها تستخدم **TCP/IP**. لا يمكنك أن تفعل شيئا للحماية ضدها إلا إذا قمت بالتبديل إلى بروتوكول آمن آخر. معظم المضادات لا تعمل إلا إذا كنت تستخدم التشفير. سرقة الهوية، وفقدان المعلومات، والاحتيال، وما الى ذلك هي المخاطر الرئيسية التي يطرحها **Session Hijacking**.

فيما يلي العناصر العرضة لـ Session Hijacking:

- **كلمات السر ذات المرة الواحدة (البطاقات الذكية، S/key، challenge response)**

جميع مخططات كلمة المرور لمرة واحدة معرضة لـ **session hijacking**. مثال على كلمة السر لمرة واحده، عند تسجل دخولك في المنتدى تتنقل بين المواضيع ولكنك لا تضطر الى اعادة تسجيل دخولك في صفحات المنتدى الاخرى لأنه تم انشاء جلسة لك بمجرد اغلاق المتصفح تتدمر الجلسة. بمجرد مصادقة المستخدم/الخدمة نفسها، فانه من الممكن أن يؤخذ الاتصال الخاص به. وفقا لموقع

www.webopedia.com فان **S/KEY** هو مخطط لكلمة المرور لمرة واحدة، وكذلك **challenge-response** المستخدمة لمصادقة الوصول إلى البيانات. الغرض من **S/KEY** هو للقضاء على الحاجة لإدخال نفس كلمة السر التي ينبغي نقلها عبر شبكة في كل مرة كان هناك الحاجة إلى كلمة المرور للوصول.

- **Kerberos**

التشفير لا يتم تمكينه افتراضيا، ونتيجة لهذا، فان الأمن هو مصدر قلق كبير لأنها تعادل مخطط كلمة المرور لمرة واحدة، والتي هي عرضة للاختطاف بسهولة.

- **Source Address Filtering Router**

الشبكة هي عرضة لهجمات عناوين الشبكة المزيفة "***network address spoof attacks***" إذا كان تأمينها يعتمد على فلترة الحزم التي تأتي من مصادر غير معروفة. فان المضيف الغير معروف يمكن إدراج نفسه، في منتصف الطريق، في اتصال موجود من قبل.

- **Source Address Controlled Proxies**

العديد من بروكسي التحكم في الوصول لبعض الأوامر تستند إلى عنوان مصدر الطالب. عنوان المصدر عرضة بسهولة لل **sniffing** سواء سلبية أو نشطة.

لم يوجد حتى الآن خطوات سهلة يمكنها تأمين الشبكة ضد **sniffing** سواء السلبي أو الإيجابي. عندما تصبح على بينة من وجود هذا التهديد، فسوف تكون أفضل استعدادا لاتخاذ قرارات أمنية ذكية لشبكة الاتصال.

لماذا Session Hijacking ناجحة

Session hijacking تكون ناجحة بسبب العوامل التالية:

- **الضعف في خوارزميات انشاء Session ID**: معظم المواقع تستخدم حاليا الخوارزميات الخطية "**linear algorithms**" بناء على متغيرات يمكن التنبؤ بها بسهولة مثل الوقت أو عنوان **IP** لتوليد **Session ID**.
  من خلال دراسة النمط المتسلسل "**sequential pattern**" وتوليد الكثير من الطلبات، فان المهاجم يمكنه بسهولة البحث لإنتاج **Session ID** صالح.
- **زمن انتهاء الجلسة "session" يكون إلى أجل غير مسمى**: معرفات الجلسات "**session IDs**" التي لها وقت انتهاء الصلاحية لأجل غير مسمى تسمح للمهاجمين مع الوقت الغير محدود من تخمين معرف جلسة "**session IDs**" صالحه. مثال على ذلك الخيار "**remember me**" الموجودة على العديد من المواقع. حيث يمكن للمهاجم استخدام معرفات الجلسة "**static-session IDs**" للوصول إلى حساب شبكة الإنترنت للمستخدم، إذا تم القبض على ملف تعريف الارتباط "**cookie file**" الخاص بالمستخدم.
  المهاجم يمكنه أيضا تنفيذ **session hijacking** إذا كان المهاجم قادرا على اقتحام خادم البروكسي، والتي يحتمل أنها تقوم بتسجيل أو تخزين معرفات الجلسة "**session IDs**".
- **نقل النص واضح**: يمكن التجسس على معرفات الجلسة "**session IDs**" عبر شبكة مسطحة بسهولة، في حال إذا لم يتم استخدام **SSL** بينما يتم نقل **session ID cookie** من وإلى المتصفح. في هذه الحالة، فإن **SSL** لا تحمي المعلومات. تصبح وظيفة المهاجم أسهل، إذا احتوى معرفات الجلسات "**session IDs**" على معلومات تسجيل الدخول الفعلية في نص واضح.
- **صغر معرفات الجلسة "session IDs"**: على الرغم من أن استخدام خوارزمية تشفير قوية، فان معرف جلسة العمل النشطة يمكن تحديده بسهولة إذا كان طول السلسلة صغيرة.
- **المعالجة الغير آمنة**: يمكن للمهاجم استرجاع المعلومات المخزنة بواسطة معرف جلسة بواسطة تضليل متصفح المستخدم لزيارة موقع آخر. ثم يمكن للمهاجم استغلال المعلومات قبل انتهاء الدورة. ويمكن تحقيق ذلك بطرق عديدة مثل **DNS poisoning**، **cross-site scripting exploitation**، أو من خلال استغلال خلل في المتصفح، الخ.
- **لا يتم تأمين الحساب لمعرفات الجلسة الغير صالحه "No Account Lockout for Invalid Session IDs"**: إذا كانت المواقع لديها أي شكل من أشكال تأمين الحساب، فان المهاجم يمكنه أن يجعل عدد من المحاولات مع اختلاف معرفات الجلسات جزءا لا يتجزأ من **URL** الحقيقي. المهاجم يمكنه أن يستمر في المحاولات حتى يتم تحديد معرف الجلسة الفعلي. هذا عادة ما يسمى **brute forcing the session IDs**. خلال هجوم **brute forcing the session IDs**، فإن خادم الويب لا يظهر على السطح أي رسالة تحذير أو شكوى. وهكذا، يمكن للمهاجم تحديد معرف الجلسة الأصلي.

جميع العوامل المذكورة أعلاه تلعب دورا هاما في نجاح session hijacking.

تقنيات Session Hijacking الرئيسية

كانت **Session Hijacking** مشكلة مستمرة لمطوري متصفح الويب وخبراء الأمن. هناك ثلاث طرق رئيسية يتم استخدامها لإجراء هجوم **Session Hijacking**:

- **Brute Forcing**

Brute forcing session IDs تنطوي على جعل الآلاف من الطلبات باستخدام جميع معرفات الجلسات المتاحة حتى يحصل النجاح للمهاجم. هذه التقنية شاملة ولكنها عملية تستغرق وقتا طويلا.

- **Stealing**

يستخدم المهاجم تقنيات مختلفة لسرقة هويات الجلسة. تقنيات مثل تركيب أحصنة طروادة على أجهزة الكمبيوتر العميلة، التجسس على حركة مرور الشبكة، **HTTP referrer header**، وهجمات **cross-site scripting attacks**.

- **Calculating**

باستخدام معرفات غير عشوائيا تم انشائها "**non-randomly generated IDs**"، حيث يحاول المهاجم حساب معرفات الجلسة. عدد من المحاولات التي تحتاج إلى القيام بها لاسترداد معرف جلسة المستخدم أو العميل يعتمد على المساحة الرئيسية "**key space**" لمعرفات الجلسة. ولذلك، فإن احتمال نجاح هذا النوع من الهجوم يمكن أن تحسب على أساس حجم والمساحة الرئيسية لمعرفات الجلسة.

Brute Forcing

يتم استخدام هجوم القوة الغاشمة "**Brute Forcing**" في الغالب من قبل المهاجمين لتخمين معرف جلسة "**session IDs**" الهدف لإطلاق الهجوم. في هذه التقنية، يحاول المهاجم إمكانيات لأنماط متعددة حتى تعمل معرف جلسة وتنجح. ويستخدم هذا الأسلوب عندما تكون الخوارزمية التي تنتج معرفات الجلسات ليست عشوائية. على سبيل المثال، في عناوين المواقع التالية، المهاجم يحاول تخمين **session ID**:

http: //www.mysite.com/view/VW30422101518909
http://www.mysite.com/view/VW30422101520803
http://www.mysite.com/view/VW30422101522507

باستخدام "**referrer attack**"، يحاول المهاجم جذب المستخدم للنقر على رابط لموقع آخر (وصلة **mysite**، على سبيل المثال، **www.mysite.com**).

GET /index.html HTTP/1.0 Host: www.mysite.com Referrer: www.mywebmail.com/viewmsg.asp?msgid=689645&SID=2556X54VA75

المهاجم يحصل على معرف جلسة المستخدم عن طريق إرسال عندما يرسل المتصفح **referrer URL** الذي يحتوي على معرف جلسة المستخدم إلى موقع المهاجم.

بعض التقنيات المستخدمة لسرقة معرفات الجلسات هي:

- **استخدام رأس HTTP referrer header.**
- **التنصت "sniffing" على حركة مرور الشبكة.**
- **استخدام هجمات cross-site scripting.**
- **إرسال أحصنة طروادة على أجهزة الكمبيوتر العميل.**

Brute Forcing Attack

المهاجم يمكنه الحصول على معرف الجلسة باستخدام أسلوب القوة الغاشمة للوصول إلى الجلسة الهدف المشروعة عندما تكون الجلسة نشطة. في هجوم "**referrer**"، المهاجم يدعو المستخدم النقر على رابط لموقع آخر. في هجمات القوة الغاشمة، يمكن للمهاجم محاولة العديد من المعرفات. على سبيل المثال، نلقي نظرة على الشكل التالي مع قائمة من عناوين المواقع، التي فيها يحاول المهاجم تخمين **session ID**.

كما تتضمن هذه التقنية تخمين معرف جلسة فإنها تحاول اختطاف الجلسة "**hijack the session**"، يجب أن يكون النطاق الممكن من قيم معرف الجلسة محدود.

ملحوظه: هجوم session ID brute forcing attack معروف بهجوم تنبأ الجلسة "session prediction attack"، إذا كان النطاق المتوقع للقيم معرف الجلسة صغير جدا.



HTTP Referrer Attack

تتبع **HTTP referrers** يمكن ان يكون فعالا في توليد الهجمات إذا تم تمرير المعلمات من خلال طلبات **GET request**. عند القيام بأي من طلب **HTTP**، فان معظم متصفحات الويب تم تكوينها لإرسال **URL** الأصلي في رأس **HTTP** يسمى **referrer**.
في هجوم **referrer attack**، المهاجم يسحر الضحية للنقر على وصلة إلى موقع ويب تحت سيطرة المهاجم. دعونا نعتبر ان موقع المهاجم سوف يكون **mysite**، على سبيل المثال، www.mysite.com.

GET /index.html HTTP/1.0 Host: www.mysite.com Referrer: www.mywebmail.com/viewmsg.asp?msgid=689645&SID=2556X54VA75

ثم يقوم متصفح الضحية بإرسال **referrer URL** يحتوي على معرف الجلسة إلى موقع المهاجم، أي **www.mysite.com**. وبما أن الموقع تحت سيطرة المهاجم، فانه يمكن بسهولة تحديد معرف الجلسة من **referrer URL**. بمجرد قيام المهاجم بتحديد معرف الجلسة، فانه يمكن أن يتخذ بسهولة الجلسة ويقوم بسرقة البيانات الحساسة للضحية.

بعض التقنيات المستخدمة لسرقة معرف الجلسة "session IDs":

- استخدام رأس **HTTP referrer header**.
- التنصت على حركة مرور الشبكة.
- باستخدام هجمات **cross-site scripting attacks**.
- إرسال أحصنة طروادة الى أجهزة كمبيوتر العميل.

Spoofing vs. Hijacking

المصدر: http://www.microsoft.com
ان اول ظهور لهجمات اختطاف الجلسة "**session hijacking attack**" ربما كانت دودة موريس التي أثرت على ما يقرب من 6،000 من أجهزة الكمبيوتر على **ARPANET** في عام 1988. وكان هذا أول حادث آلي على أمن الشبكات **ARPANET** حيث قام روبرت موريس بكتابة برنامج يمكن أن ينتشر من خلال عدد من أجهزة الكمبيوتر وتواصل عملها في حلقة لا نهائية، في كل مرة يقوم

بنسخ نفسه إلى جهاز كمبيوتر جديد على **ARPANET**. واستند العمل الأساسي للدودة موريس على اكتشاف أمن اتصال **TCP/IP** القائم على تسلسل الأرقام، وأنه كان من الممكن التنبؤ بها.

Blind hijacking ينطوي على توقع أرقام التسلسل "**sequence numbers**" التي يرسلها المضيف الهدف من أجل إنشاء اتصال يبدو أنه صادر من المضيف. قبل الاستكشاف بالتحايل الأعمى "**blind spoofing**"، نلقي نظرة على التنبؤ رقم التسلسل. أرقام تسلسل **TCP**، والتي هي فريدة من نوعها لكل بايت في جلسة **TCP**، توفر التحكم في التدفق وسلامة البيانات لنفسه. وبالإضافة إلى ذلك، جزء **TCP** يعطي رقم تسلسل أولي (**ISN**) كجزء من رأس القطاع "**segment header**". لا يبدأ رقم التسلسل الأولي "**ISN**" عند المستوى صفر لكل جلسة. يتم ترقيم **ISN** كجزء من عملية المصافحة في اتجاهات مختلفة، والبايت بالتتابع. يعتمد **Blind IP hijacking** على قدرة المهاجم على التنبؤ بأرقام التسلسل، لأنه إذا كان غير قادر على التنصت على التواصل بين اثنين من المضيفين بحكم انه ليس على نفس جزء الشبكة. المهاجم لا يمكنه تزوير مضيف موثوق به "**spoof host**" على شبكة مختلفة ورؤية حزم الرد "**reply packets**" لأن الحزم لا يتم إعادة توجيهها له. لا يمكنه أيضا **ARP cache poisoning** بسبب ان الراوتر لا يمكنه بث **ARP** عبر الإنترنت. كما ان المهاجم غير قادر على رؤية الردود، فانه يضطر الى استباق الردود من الهدف ومنع المضيف من إرسال **RST** إلى الهدف. المهاجم يقحم نفسه في الاتصال من خلال التنبؤ بأرقام تسلسل المضيف البعيد. ويستخدم هذا على نطاق واسع لاستغلال علاقات الثقة بين المستخدمين والأجهزة البعيدة. وتشمل هذه الخدمات **NFS**، **telnet**، و**IRC**.

IP spoofing من السهل تحقيقه. لخلق حزم خام جديدة، والشرط الوحيد هو أن المهاجم يجب أن يكون له الوصول الجذري على الجهاز. من أجل تأسيس اتصال مزيف، يجب على المهاجم ان يعرفوا ما هي أرقام التسلسل التي يتم استخدامها. ولذلك، **IP spoofing** يجبر المهاجم للتنبؤ برقم التسلسل المقبل. لإرسال الأوامر، يستخدم المهاجم **blind hijacking**، ولكن الرد لا يمكن النظر له.

- في حالة **IP spoofing**، تخمين رقم التسلسل غير مطلوب لأنه ليس هناك أي جلسة مفتوحة حاليا مع عنوان **IP**. في **blind hijacking**، فإن حركة المرور يجب ان تعود إلى المهاجم باستخدام توجيه المصدر الوحيد. هذا هو المكان الذي فيه يخبر المهاجم الشبكة بكيفية توجيه المخرجات والمدخلات من الجلسة، وانه يقوم بـ **promiscuously sniffs** من الشبكة لأنها تمر من قبل المهاجم. وتستخدم أوراق اعتماد المصادقة "**authentication credentials**" الملتقطة لتأسيس جلسة عمل في الجلسة المزيفة "**session spoofing**". هنا، **active hijacking** يحجب جلسة موجودة من قبل. ونتيجة لهذا الهجوم، يمكن للمستخدم الشرعي فقد الوصول أو يمكن أن يكونوا محرومين من وظائف عادية من جلسة عمل **Telnet** له والتي تم اختطافها من قبل المهاجم، الذي يعمل الآن مع امتيازات المستخدم. وبما أن معظم المصادقة تحدث عندما يتم بدا الجلسة، وهذا يسمح للمهاجمين للوصول إلى الجهاز الهدف. طريقة أخرى هي استخدام حزم **source-routed IP packets**. وهذا يسمح للمهاجمين ليصبحوا جزءا من محادثة المضيف المستهدف عن طريق خداع توجيه حزم **IP** بالمرور عبر النظام الخاص به.
- **Session hijacking** هو أكثر صعوبة من **IP address spoofing**. على سبيل المثال في **session hijacking**، فأن جون (المتسلل) يسعى إلى إدراج نفسه في جلسة جين (مستخدم الشرعي) بالفعل قد وضعت مع **//Mail**. جون ينتظر حتى يتم تأسيس الجلسة، ثم يقوم بضربها من قبل بعض الوسائل والتقاط الجلسة وكأنه جين. ثم يقوم جون بإرسال مجموعة حزم تم كتابتها "**scripted packet**" إلى **//Mail**، وسوف يكون قادرا على رؤية الردود. للقيام بذلك، فانه بحاجة لمعرفة رقم التسلسل لاستخدامه عند خطف الجلسة، والتي يمكن أن تحسب نتيجة معرفة **ISN** وعدد الحزم التي تم تبادلها.
- دورة **Session hijacking** الناجحة تكون صعبه من دون استخدام أدوات معروفة وممكن فقط عندما يكون عدد من العوامل تحت سيطرة المهاجم. معرفة **ISN** يكون أقل شيء في تحديات جون. على سبيل المثال، فإنه بحاجة إلى وسيلة لضرب جين عندما يريد، وأيضا بحاجة إلى وسيلة لمعرفة الوضع الدقيق لجلسة جين في اللحظة التي شنا هجومه. كل هذه يتطلب أن جون يكون لديه أكثر بكثير من المعرفة والسيطرة على الجلسة.
- مع ذلك، هجمات **IP address spoofing attacks** يمكن ان تتكلل بالنجاح إذا تم استخدام عناوين **IP** للمصادقة. المهاجم لا يمكنه تنفيذ **IP address spoofing** أو **Session hijacking** إذا تم تنفيذ فحص سلامة كل حزمة. على نفس الطريق، **IP address spoofing** و**session hijacking** ليست ممكنة إذا استخدمت جلسة مشفره مثل **SSL** أو **PPTP**. ونتيجة لذلك، فان المهاجم لا يمكنه أن يشارك في تبادل المفاتيح.
- باختصار، اختطاف اتصالات **TCP** غير مشفرة يتطلب وجود حركة مرور لجلسه غير مشفره، القدرة على التعرف على أرقام تسلسل **TCP** والتي تتنبأ برقم تسلسل (**NSN**) التالي، والقدرة على محاكاة عنوان **MAC** المضيف أو عنوان **IP** من أجل تلقي الاتصالات التي لم يتم توجيهها للمضيف المهاجم. إذا كان المهاجم موجود على القطعة المحلية، فأنه يكون قادر على **sniffing** والتنبؤ برقم **ISN+1** وتوجيه حركة المرور إليه من قبل تسميم **ARP cache** على اثنين من المضيفين الشرعيين المشاركين في الجلسة.



عملية اختطاف الجلسة "Session Hijacking Process"

انه من الأسهل التسلل للدخول كمستخدم حقيقي بدلا من أن يدخل النظام مباشرة. اختطاف الجلسة "**session hijacking**" يعمل من خلال إيجاد جلسة أنشئت والاستيلاء على تلك الجلسة بعد ان يكون هناك مستخدم حقيقي لديه حق الوصول وتم المصادقة. بمجرد اختطاف او الاستيلاء على الجلسة، فان المهاجم يمكنه البقاء على اتصال لمدة ساعات. وهذا يترك متسعا من الوقت للمهاجمين لزراعة **backdoor** أو حتى اكتساب وصول إضافي لهذا النظام. واحدة من الأسباب الرئيسية التي جعلت اكتشاف اختطاف الجلسة معقد هي ان المهاجم يظهر بهوية مستخدم حقيقي. لذلك، كل حركة المرور يتم توجيهها إلى عنوان **IP** الخاص بالمستخدم والذي يأتي إلى نظام المهاجم.

كيف يمكن للمهاجمين القيام باختطاف الجلسة؟ **session hijacking** يمكن تقسيمها إلى ثلاثة مراحل رئيسية هي:

- **تتبع الاتصال "Tracking the connection":**

المهاجم ينتظر لإيجاد هدف مناسب ومضيف باستخدام **network sniffer** لتعقب الهدف والمضيف، أو لتحديد مستخدم مناسب عن طريق الفحص مع أداة مثل **NMAP** للعثور على الهدف مع وسيلة سهلة للتنبؤ بتسلسل **TCP**. وهذا للتأكد من أن ارقام التسلسل والإقرار "**acknowledgement**" التي يتم التقاطها صحيحه، حيث يتم فحص الحزم من قبل **TCP** من خلال ارقام التسلسل و/ أو الإقرار. يستخدم المهاجم هذه الأرقام لبناء الحزم.

- **إعادة مزامنة الاتصال "Desynchronizing the connection":**

حالة **desynchronized state** هو عندما يكون الاتصال بين الهدف والمضيف قائم؛ أو في حالة مستقرة بدون نقل أي من البيانات. أو رقم التسلسل الملقم لا يساوي رقم إقرار "**ACK No.**" العميل. أو رقم تسلسل العميل لا يساوي رقم إقرار "**ACK No.**" الخادم.
لعمل **desynchronize** للاتصال بين الهدف والمضيف، فان رقم التسلسل أو رقم الإقرار (**SEQ/ACK**) للملقم يجب تغييره. ويتم ذلك عن طريق إرسال بيانات فارغة إلى الملقم بحيث يمكن لأرقام **SEQ/ACK** للملقم ان تتقدم في حين أن الجهاز الهدف لا يمكنه تسجيل مثل هذه الزيادة. على سبيل المثال، قبل **desynchronization**، المهاجم يراقب الجلسة بدون أي نوع من التدخل. المهاجم يرسل كمية كبيرة من "البيانات الخالية" إلى الملقم. تخدم هذه البيانات فقط لتغيير عدد **ACK** على الخادم ولا يؤثر على أي شيء آخر. والآن، كلا من الملقم والهدف أصبحوا **desynchronized.**
ثمة نهج آخر هو أن ترسل **reset flag** إلى الملقم من أجل اسقاط الاتصال على جانب الملقم. ومن الناحية المثالية، يحدث هذا في مرحلة الإعداد الأولى من الاتصال. هدف المهاجم هو قطع الاتصال على جانب الملقم وإنشاء واحدة جديدة مع عدد تسلسل مختلف.
المهاجم يصغي لحزمة **SYN/ACK** من الخادم إلى المضيف. وبمجرد الكشف عن الحزمة، فان المهاجم على الفور يرسل حزمة **RST** إلى الملقم وحزمة **SYN** مع ضبط نفس المعلمات، مثل رقم المنفذ، ولكن مع رقم تسلسل مختلف. الخادم، بعد تلقي حزمة **RST**، فانه يقوم

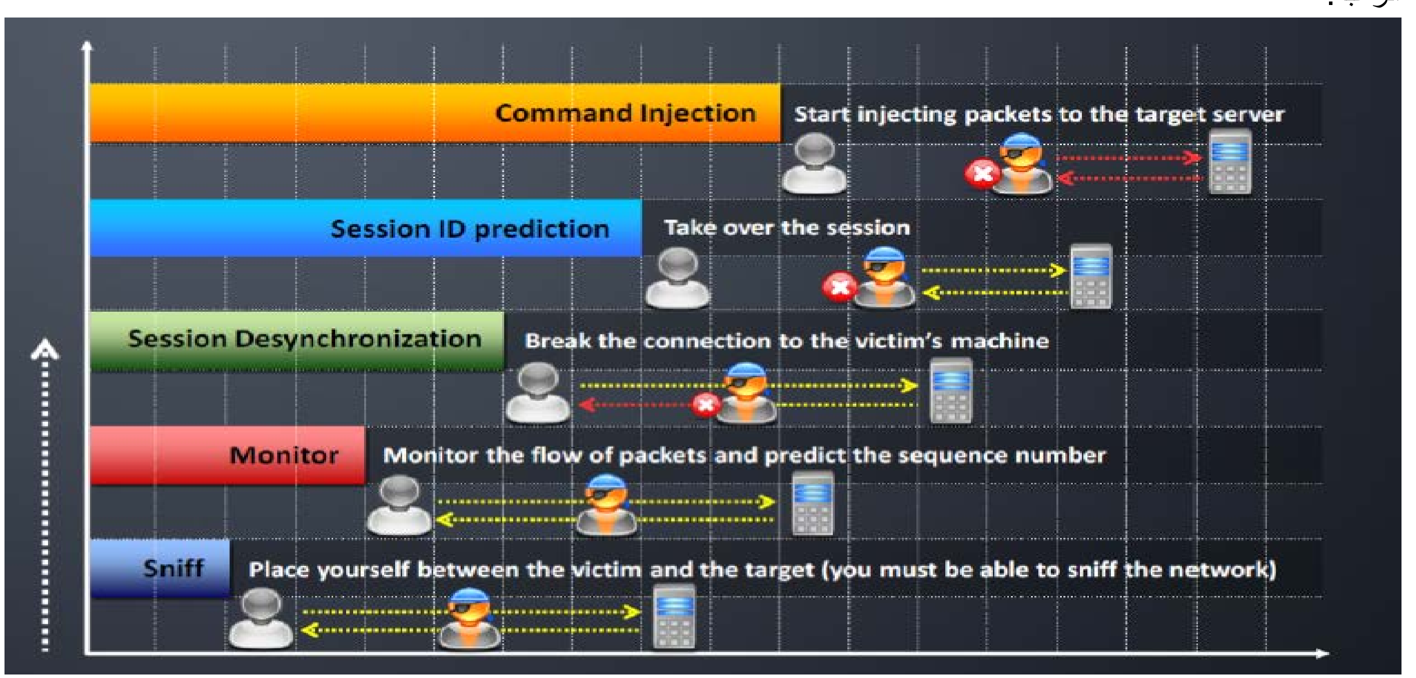
بإغلاق الاتصال مع الهدف ويبدأ اتصال معه على أساس حزمة **SYN**، ولكن مع رقم تسلسل مختلفة على نفس المنفذ. بعد فتح اتصال جديد، الملقم يرسل حزمة **SYN/ACK** إلى الهدف من اجل الإقرار "**acknowledgement**". المهاجم يقوم بالكشف (ولكنه لا يعترض) هذا ويرسل إلى حزمة **ACK** إلى الملقم. الآن الملقم في حالة **established state**. الهدف الرئيسي هو الحفاظ على الهدف ملما، والتحول إلى حالة **established state** عندما يتلقى أول حزمة **SYN/ACK** من الخادم. الآن كلا من الخادم والهدف هم في حالة **desynchronized**، ولكن في حالة **established state**.

يمكن أيضا أن يتم ذلك باستخدام **FIN flag**، ولكن يمكن أن يسبب هذا ان يقوم الخادم بالرد مع **ACK** ومن ثم ابعاد الهجوم من خلال عاصفة **ACK**. يحدث هذا بسبب وجود عيب في هذه الطريقة من **hijacking a TCP connection**. في حين تلقي حزمة غير مقبولة، فان المضيف يقر ذلك عن طريق إرسال رقم التسلسل المتوقع. هذه الحزمة غير مقبولة يولد حزمة الإقرار، وبالتالي خلق حلقة لا نهاية لكل حزم البيانات. عدم التوافق في ارقام **SEQ/ACK** ينتج عنه حركة مرور زائده لشبكة كل من الخادم والهدف حيث يحاولا التحقق من التسلسل الصحيح. هذه الحزم لا تحمل بيانات، ولا يتم إعادة إرسالها أنهم في حالة فقدانها. ومع ذلك، منذ استخدام **TCP** لـ **IP**، فان فقدان حزمة واحدة يضع نهاية للمحادثة غير مرغوب فيها بين الخادم والهدف.

مرحلة **desynchronizing** تم اضافتها في **hijack sequence**، لذا المضيف الهدف جاهل عن هذا الهجوم. بدون **desynchronizing**، المهاجم قادر على ضخ البيانات إلى الخادم ويبقي هويته عن طريق خداع عنوان **IP**. ومع ذلك، فانه لديه طرح مع استجابة الملقم ليتم ترحيلها إلى المضيف الهدف أيضا.

- **حقن حزمة المهاجم "Injecting the attacker's packet":**

الآن بعد أن قام المهاجم بقطع الاتصال بين الخادم والهدف، فانه يمكن أن يختار إما لحقن البيانات في الشبكة أو المشاركة بنشاط مثل رجل في المنتصف "**man-in-the-middle**"، تمرير البيانات من الهدف الى الخادم، والعكس بالعكس، والقراءة عن طريق حقن البيانات حسب الرغبة.



Packet Analysis of a Local Session Hijack

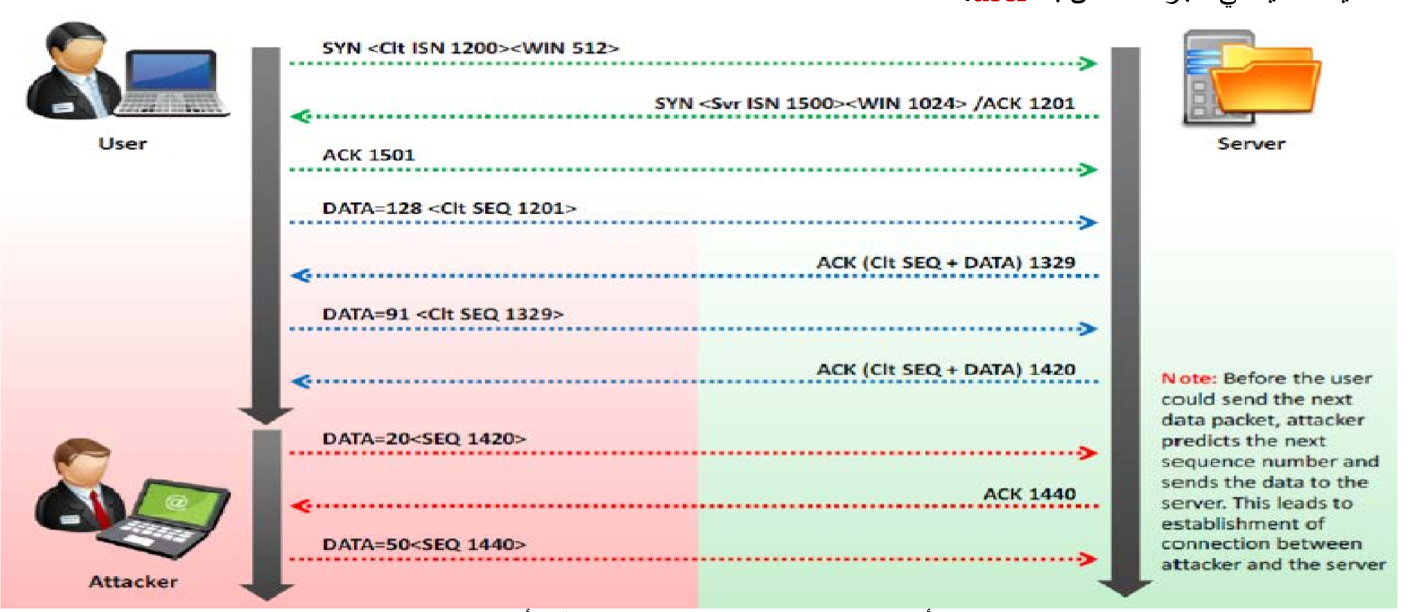
هجمات **Session hijacking** هي ناقلات هجوم رفيعة المستوى التي تؤثر على العديد من النظم. العديد من الأنظمة التي ترتبط في **LAN** أو تستخدم بروتوكول اتصال الإنترنت **TCP** لنقل البيانات. لإنشاء اتصال بين نظامين ولنقل ناجح من البيانات، يجب على النظامين إنشاء المصافحة الثلاثية. جلسة الاختطاف "**Session hijacking**" ينطوي على استغلال نقطا ضعف طريقة المصافحة الثلاثية "**three-way handshake**" للسيطرة على الجلسة.

لإجراء هجوم خطف الجلسة "Session hijacking"، فان المهاجم يقوم بثلاثة من الأنشطة:

- **تتبع الجلسة "tracks the session".**

- .Desynchronizes the session
- حقن أوامر المهاجم "injects attacker's commands".

يمكن رصد جلسة أو تعقبها ببساطة عن طريق التجسس على حركة المرور. المهمة التالية في اختطاف الجلسة هي **desynchronize**. ويمكن تحقيق ذلك بسهولة إذا كان عدد التسلسل المقبل الذي سوف يتم استخدامه العميل معروف. إذا كنت تعرف رقم التسلسل، فانه يمكنك اختطاف الجلسة باستخدام رقم التسلسل قبل العميل. هناك احتمالان لتحديد أرقام التسلسل. احدى الطرق هو التنصت على حركة المرور، وإيجاد حزمة **ACK** ومن ثم تحديد رقم التسلسل المقبل على أساس حزمة **ACK**. والطريقة الأخرى هي نقل البيانات مع تخمين أرقام التسلسل. والطريقة الثانية ليست موثوقة جدا. إذا كنت تستطيع الوصول إلى الشبكة، فيمكن التنصت على جلسة **TCP**، ثم يمكنك تحديد رقم التسلسل بسهولة. ويسمى هذا النوع من اختطاف الجلسة "**local session hijacking**". وفيما يلي تحليل لحزمة **TCP** ذات المصافحة الثلاثية العادية في الجزء الخاص بال **user**.



بناء على هذا الرسم، فإن رقم التسلسل المتوقع أن يكون القادم 1420. إذا كنت تستطيع أن ترسل حزمه ذات رقم التسلسل هذا قبل المستخدم، فانه يمكنك **desynchronize** الاتصال بين المستخدم والخادم. من خلال الرسم البياني السابق في الجزء الخاص بال **Attacker** فانه يبين تحليل لحزمة من **local session hijacking**.

المهاجم يرسل البيانات مع رقم تسلسل متوقع قبل ان يرسل المستخدم ذلك. الآن، سوف يكون الملقم في التزامن مع المهاجم. وهذا يؤدي إلي إنشاء علاقة بين المهاجم والملقم. بمجرد ان يتم تأسيس الاتصال بين المهاجم والملقم، على الرغم من أن المستخدم يرسل البيانات مع رقم التسلسل الصحيح، فان الملقم يسقط البيانات معتبرة تلك الحزمة بالاستياء. المستخدم يجهل عمل المهاجم ويمكن إعادة إرسال حزم البيانات كما هي حيث انه لا يتلقى **ACK** لحزمة **TCP** له. ومع ذلك، يسقط الملقم الحزمة مرة أخرى. وهكذا، المهاجم ينفذ هجوما اختطاف الجلسة المحلي.

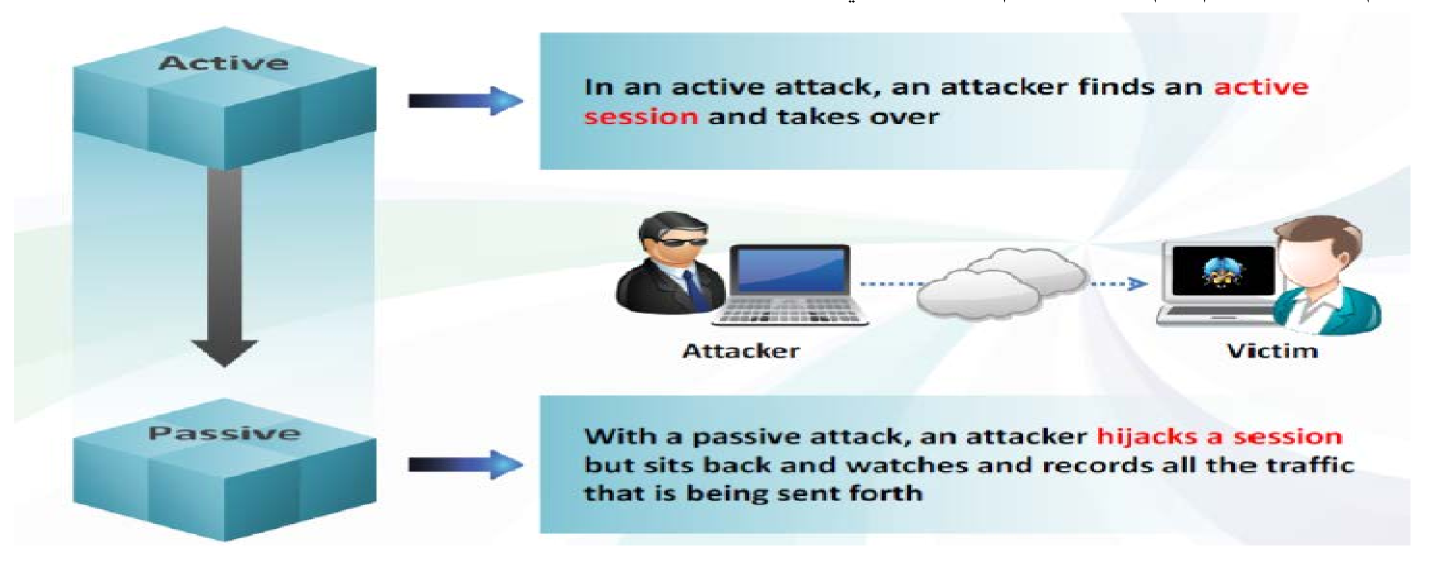
أنواع اختطاف الجلسة "Types Of Session Hijacking"

جلسة اختطاف "Session hijacking" يمكن أن تكون إما إيجابية أو سلبيه، اعتمادا على درجة مشاركة المهاجم. الفارق الجوهري بين الايجابي والسلبي هو أنه في حين أن الخطف الإيجابي يأخذ جلسة عمل موجودة، اما الخطف السلبي تراقب الجلسة الجارية.

يستخدم الهجوم السلبي التجسس "**sniffing**" على الشبكة مما يسمح للمهاجمين للحصول على معلومات مثل معرفات المستخدمين وكلمات المرور. يمكن للمهاجم بعد ذلك استخدام هذه المعلومات لتسجيل الدخول كمستخدم صالح والاستيلاء على الامتيازات. التنصت على كلمات المرور "**password sniffing**" هو هجوم بسيط يمكن القيام به عندما يتم الحصول على وصول الى الشبكة. لمواجهة هذا الهجوم عن طريق الأساليب التي تتراوح من مخططات التحديد (مثل كلمة المرور لمرة واحدة مثل **skey**) الى تحديد التذاكر (مثل كيربيروس). هذه التقنيات لحماية البيانات من ان يتم التنصت عليها، لكنهم لا يستطيعون حمايته من الهجمات الفعالة ما لم يتم تشفيرها أو حمل توقيع رقمي.

في الهجوم الفعال، المهاجم يأخذ أكثر من جلسة عمل موجودة إما عن طريق هدم الاتصال على جانب واحد من المحادثة، أو من خلال المشاركة بنشاط كرجل في المنتصف **MITM**. مثال على هجوم نشط هو هجوم **MITM**. لتحقيق النجاح لهذا النوع من الهجوم، يجب أن

تخمن رقم التسلسل قبل ان يستجيب الهدف إلى الخادم. في الوقت الحاضر، التنبؤ بأرقام التسلسل لم يعد صالحا لتنفيذ هجوم ناجح لأن بائعي نظام التشغيل تستخدم القيم العشوائية لرقم التسلسل الأولي.



Session Hijacking in the OSI Model

Session hijacking in the OSI model يمكن إجراءها على مستويين، مستوى الشبكة ومستوى التطبيق. الاختطاف على مستوى الشبكة "**Network-level hijacking**" يمكن تعريفها بأنها فعل اختراق جلسة **TCP** و**UDP** بين العميل والخادم وثم اعتراض الحزم أثناء نقل البيانات. في الاختطاف على مستوى الشبكة "**Network-level hijacking**"، المهاجم يقوم بجمع المعلومات الحاسمة التي يمكن استخدامها لشن هجوم على مستوى التطبيق. في الاختطاف على مستوى التطبيق "**application-level hijacking**"، المهاجم يقوم بالاعتراض في تطبيق ويب.

الاختطاف على مستوى التطبيق "**application-level hijacking**" هو عباره عن السيطرة على جلسة **HTTP** المستخدم من خلال الحصول على معرفات الجلسة. هنا يتم الهجوم على الجلسة الحالية والمهاجم يمكنه توليد جلسات عمل جديدة استنادا إلى المعلومات المسروقة.

معرفات الجلسات "Session IDs" يمكن العثور عليها في:

- مدمجة في عنوان **URL**، التي تم استقبالها بواسطة التطبيق من اجل **GET request**.
- في الحقول المخفية من **form**.
- في ملفات الكوكيز التي يتم تخزينها في الجهاز المحلي للعميل.



11.2 اختطاف الجلسة على مستوى التطبيق (Application Level Session Hijacking)

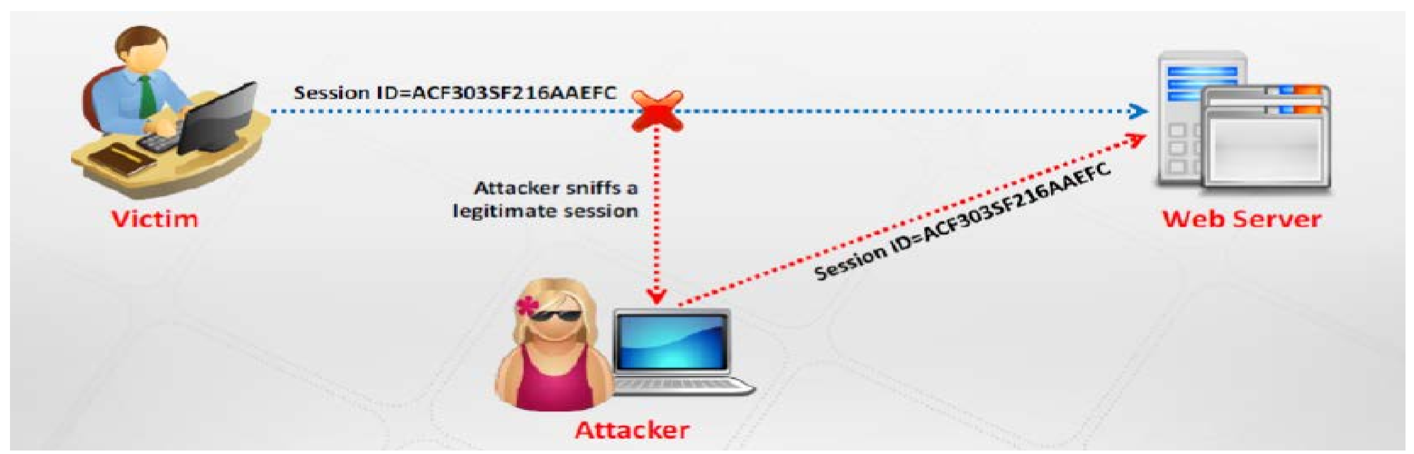
حتى الآن، قد ناقشنا مفاهيم مختلفة من اختطاف الجلسة، وأنواع اختطاف الجلسة، واختطاف الجلسة في نموذج **OSI**. الآن سوف نناقش اختطاف الجلسة على مستوى التطبيق، و **level of hijacking** في نموذج **OSI**. يصف هذا القسم مفهوم اختطاف الجلسة على مستوى التطبيق ومختلف التقنيات المستخدمة لتنفيذ ذلك.

في هجوم اختطاف الجلسة "**Session Hijacking**"، فان رموز الجلسة "**session token**" تكون معرض للاختراق أما عن طريق التنبؤ أو سرقة رمز جلسة صالح لكسب امتيازات غير مصرح بها على خادم الويب. وكما ذكر سابقا، فان الاختطاف على مستوى الشبكة يوفر المعلومات المفيدة التي يمكن استخدامها لأداء الاختطاف على مستوى التطبيق. وبالتالي، الاختطاف على مستوى الشبكة وعلى مستوى التطبيق تحدث معا في معظم الحالات. الاختطاف على مستوى التطبيق "**Application level session hijacking**" ينطوي إما للسيطرة على جلسة عمل موجودة أو إنشاء جلسة جديدة استنادا إلى البيانات المسروقة. يحدث الاختطاف على مستوى التطبيق مع جلسات **HTTP Session**. جلسات **HTTP Session** يمكن خطفها/الاستيلاء عليها عن طريق الحصول على معرفات الجلسة "**Session IDs**" منها، وهي معرفات فريدة. وهناك الطرق المختلفة لاختطاف الجلسة على مستوى التطبيق ويمكن تحقيق ذلك عن طريق اختراق رمز الجلسة على النحو التالي:

- **التنبؤ برمز الجلسة "Predictable session token".**
- **هجوم رجل في الوسط "Man-in-the-middle attacks".**
- **هجمات على جهاز العميل مثل (XSS, malicious JavaScript Codes, Trojans, etc.).**
- **هجمات Man-in-the-browser attacks.**
- **التجسس على الجلسة "session sniffing".**

التجسس على الجلسة "session sniffing"

التنصت على الجلسة "**session sniffing**" من السهل جدا القيام بها إذا تم إرسال حركة مرور **HTTP** غير مشفرة. قد تحتوي على حركة مرور **HTTP** على معرفات الجلسة. المهاجمين يقومون باستخدام **sniffers** لالتقاط حركة مرور **HTTP** ومن ثم تحليل الحزم لتحديد هويات الجلسة. يمكن للمهاجمين تحديد معرفات الجلسة بسهولة حيث ان حركة المرور غير مشفرة. وقد تحتوي الجلسة الغير مشفرة أيضا على معلومات حول أسماء المستخدمين وكلمات المرور. يوضح الشكل التالي شرح بياني لكيفية تنصت المهاجم على الجلسة:



في البداية يقوم المهاجم بالتنصت على حركة مرور **HTTP** بين الضحية وخادم الويب ويحلل البيانات التي تم التقاطها ويحدد **ID** الجلسة. ثم، يقوم المهاجم بتزييف نفسه بأنه الضحية ويرسل **ID** الجلسة إلى خادم الويب قبل الضحية. وهكذا، يأخذ المهاجم السيطرة على جلسة عمل موجودة.

التنبؤ برمز الجلسة "Predictable session token"

توقع رموز الجلسة (معرفات الجلسات "**session ID**") هو وسيلة لاختطاف أو انتحال صفة مستخدم الموقع. وهو معروف أيضا باسم اختطاف الجلسة "**session hijacking**" أو طريقة التنبؤ بالجلسة/أوراق الاعتماد "**session/credential prediction method**". ويمكن

تحقيق ذلك عن طريق التخمين أو بناء قيمة فريدة، على سبيل المثال معرف الجلسة "**session ID**" والتي تستخدم لتحديد هوية المستخدم أو جلسة معينة. باستخدام تقنية اختطاف الجلسة، فان المهاجم يكون لديه القدرة على القيام بطلبات **ping** على موقع ما مع امتيازات المستخدم المخترق.

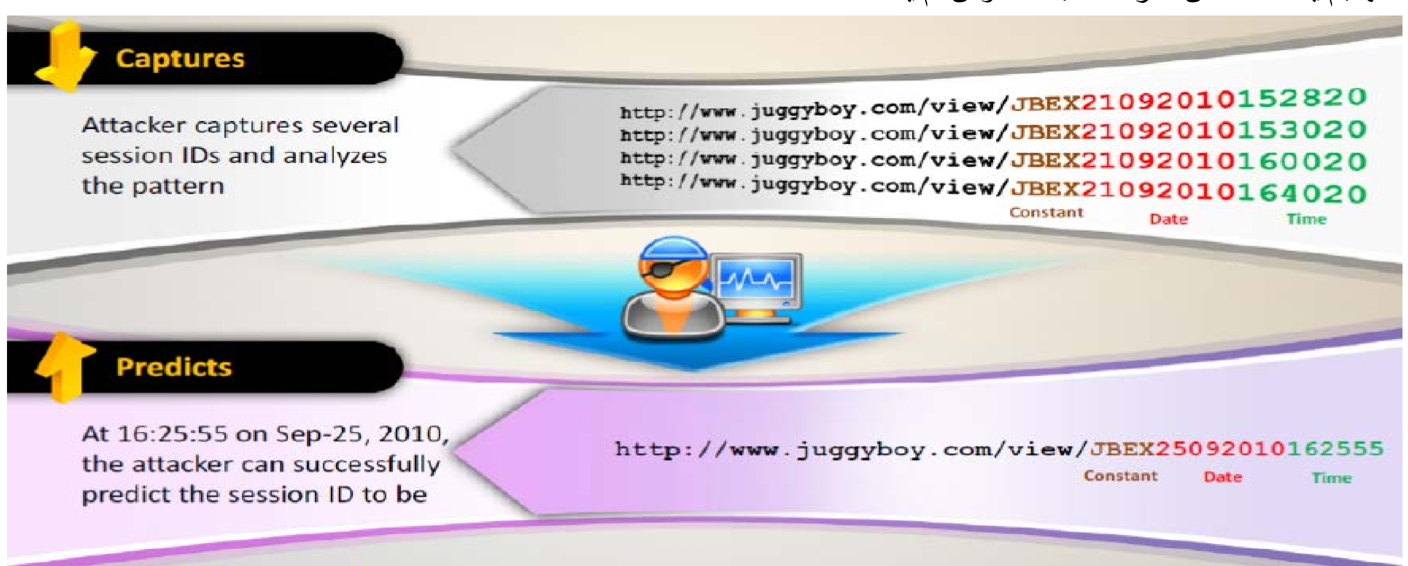
عندما يرسل المستخدم طلب لموقع على شبكة الانترنت للاتصال، يحاول الموقع لأول مرة توثيق وتتبع هوية المستخدم. ما لم يثبت هوية المستخدم، فإن الموقع لا يوفر المعلومات المطلوبة للمستخدم. المواقع عادة تقوم بعملية مصادقة المستخدم على أساس مزيج من اسم المستخدم وكلمة المرور (**credentials**). عندما يقوم المستخدم بتسجيل اسم المستخدم وكلمة المرور، فان الموقع يولد "معرف جلسة" فريد. معرف الجلسة هذا يشير الى جلسة المستخدم. معرف الجلسة "**session ID**" هي معلمات لاحقة بالاتصالات بين المستخدم والموقع كدليل على جلسة المصادقة. إذا كان المهاجم قادرا على تحديد معرف الجلسة هذا إما من خلال التنبؤ أو التخمين، فأنه قادر على اختراق جلسة عمل المستخدم.

كيف يمكن التنبؤ برمز الجلسة "How to Predict a Session Token"؟

معظم خوادم الويب تستخدم خوارزميات مخصصه أو نمط محدد مسبقا لتوليد معرفات الجلسات "**sessions IDs**". الخوارزميات قد تولد معرفات الجلسات عن طريق زيادة أعداد ثابتة أو باستخدام إجراءات معقدة مثل **factoring in time** أو متغيرات الكمبيوتر الأخرى المحددة. بمجرد ان يتم احتساب معرف الجلسة، يتم تخزينها في **URL**، في حقل نموذج مخفي "**hidden form field**"، أو في ملف الكوكيز. في مثل هذه الحالات، يمكن للمهاجم بسهولة تحديد معرف الجلسة، إذا تمكن من تحديد الخوارزمية المستخدمة لتوليد معرفات الجلسات "**sessions IDs**". الطرق الممكنة التي يمكن من خلالها ان يطلق المهاجمين هجماتهم تكون على النحو التالي:

- الاتصال بتطبيق الويب للحصول على معرف الجلسة.
- استخدام **Brute force** أو حساب معرف جلسة المقبل.
- تبديل القيمة الحالية في **URL/hidden form-field/cookie** بالتالي يمكن افتراض هوية المستخدم القادمة.

المهاجم يلتقط عدد من معرفات الجلسات ومن ثم يحلل النمط



هجوم رجل في الوسط "Man-In-The-Middle-Attacks"

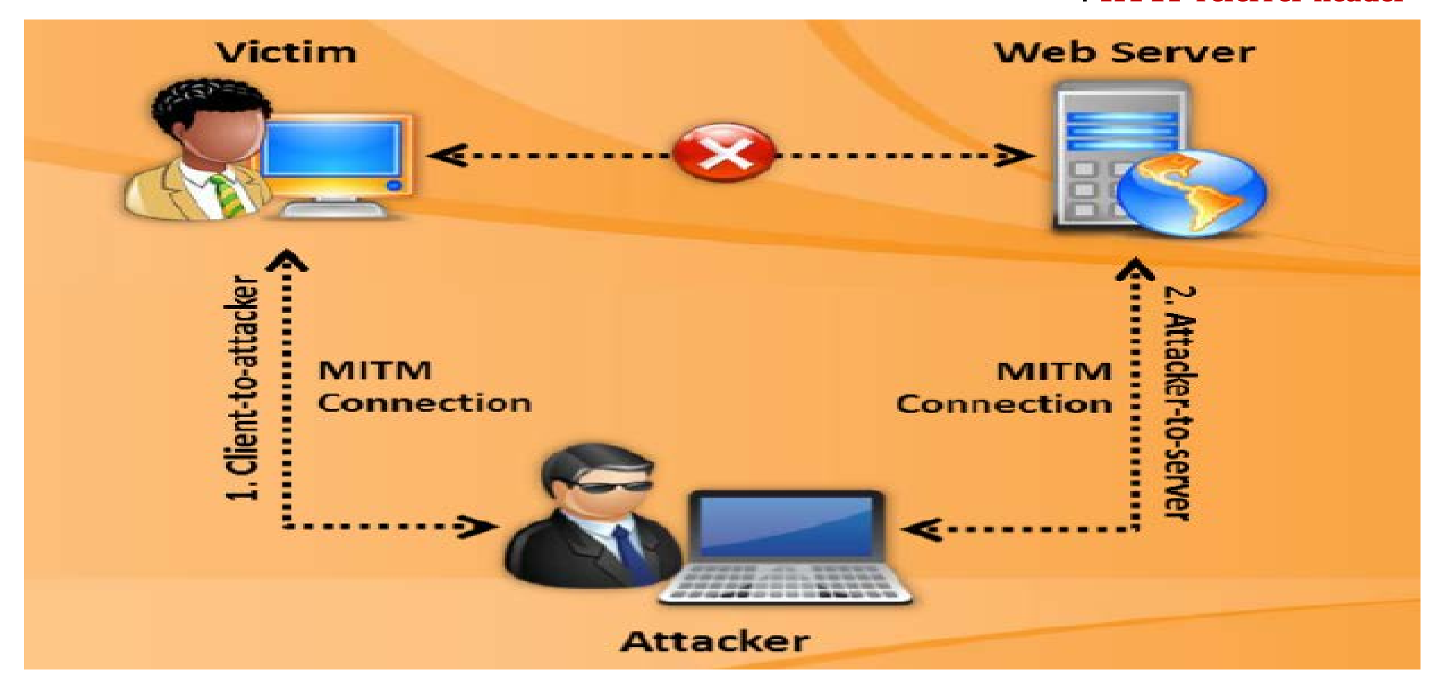
هجوم رجل-في-الوسط هو نوع من الهجوم الذي يتدخل فيه المهاجمين في اتصال موجود بين نظامين لاعتراض الرسائل التي يتم تبادلها ولضخ معلومات احتيالية. الضحية هنا تعتقد انها تتحدث مباشرة مع شخص آخر، ولكن في واقع الأمر يتم التحكم في المحادثة بأكملها من قبل المهاجم. ويشمل هذا الهجوم وظائف مختلفة من التطفل على اتصال، التداخل في الاتصال، اعتراض الرسائل، وتعديل البيانات.

دعونا نتأمل في المثال في معاملة **HTTP**. في هذه الحالة، الهدف هو اتصال **TCP** بين العميل والخادم. المهاجم يقوم بشق اتصال TCP شرعي بين العميل والخادم في اتصالين باستخدام تقنيات مختلفة. هذين الاتصالين هما:

- اتصال بين المهاجم والعميل.
- اتصال بين المهاجم والخادم/الملقم.

بعد الاعتراض الناجح لاتصالTCP ، فان للمهاجم يمكنه قراءة وتعديل وادخال بيانات غير صحيحة في الاتصالات التي اعتراضها.

ونظرا لطبيعة بروتوكول **HTTP** ونقل البيانات التي تعتبر كلها قائمه على **ASCII**، فان هجوم رجل في الوسط فعال. وبهذه الطريقة، من الممكن عرض البيانات المنقولة عبر بروتوكول **HTTP** وأيضا من الممكن التقاط معرف جلسة عمل من خلال قراءة رأس **HTTP** "**HTTP referrer header**".



هجوم Man-in-the-Browser Attacks

هجوم **man-in-the-browser attack** مشابه لحد كبير هجوم رجل في الوسط **MITM**. الفرق بين الطريقتين هو أن هجوم **man-in-the-browser attack** يستخدم حصان طروادة لاعتراض والتلاعب بالمكالمات بين المتصفح وآليات أمنها أو المكتبات. هذا الهجوم يستخدم بالفعل تثبيت طروادة على نظام العمل بين المتصفح وآليات أمنها. وهذا الهجوم قادر على تعديل والتجسس على المعاملات. الهدف الرئيسي من هذا الهجوم هو السرقة المالية عن طريق التلاعب في معاملات أنظمة الخدمات المصرفية عبر الإنترنت. مع هذه التقنية، فإن المهاجمين قادرين على سرقة المعلومات الحساسة أو المال دون أن تترك أي نوع من الإثبات أو أن يلحظ ذلك، على الرغم من أنه يتم تعيين مستوى أمان للمتصفح مرتفع. لن تجد أي إشارة عن هذا النوع من الهجمات، حتى عندما تتم المعاملات المصرفية الصافية عبر قناة **SSL**. جميع الآليات الأمنية تعرض انها تعمل بشكل طبيعي. لذلك، يجب أن يكون المستخدم ذكي ومنتبه عند استخدام أنظمة الخدمات المصرفية عبر الإنترنت.

الخطوات المتبعة لإجراء هجوم man-in-the-browser

من أجل أداء هجوم **man-in-the-browser** ناجح، يجب على المهاجم تنفيذ الخطوات التالية:

- الخطوة 1: أولا يقوم المهاجم بإصابة برمجيات الكمبيوتر (نظام التشغيل او التطبيقات على نظام التشغيل) بالتروجان.
- الخطوة 2: بعد قيام المستخدم بإعادة تشغيل المتصفح، فانه يتم تحميل الأكواد الخبيثة الموجودة في شكل ***extension files***.
- الخطوة 3: عند تحميل الصفحة، يستخدم **extension** عنوان **URL** ومن ثم مقارنته بقائمة من المواقع المعروفة المستهدفة للهجوم.
- الخطوة 4: يقوم بتسجيل **button event handler** عندما يتم الكشف عن تحميل الصفحة المحددة لنمط معين ويقارن ذلك مع قائمة المستهدفين به.
- الخطوة 5: يقوم التروجان بتثبيت الشيفرات الخبيثة (**extension files**) وحفظه في اعدادات المستعرض.
- الخطوة 6: **extension files** تقوم بتسجيل **handler** لكل زيارة للصفحة الويب.
- الخطوة 7: تسجيل دخول المستخدم الآمن إلى الموقع.
- الخطوة 8: المتصفح يرسل قيم النموذج وتعديلها إلى الملقم.
- الخطوة 9: عندما ينقر المستخدم على الزر "**button**"، فان **extension** يستخدم واجهة **DOM** ويقوم باستخراج كافة البيانات من جميع حقول النموذج وتعديل القيم.

- o الخطوة 10: بعد تنفيذ الملقم الصفقة، يتم إنشاء الاستلام.
- o الخطوة 11: المتصفح يعرض الاستلام مع التفاصيل الأصلية.
- o الخطوة 12: يتلقى الخادم القيم المعدلة ولكن لا يمكنه التمييز بين الأصلي والقيم المعدلة.
- o الخطوة 13: الآن، المتصفح يتلقى استلام لهذه الصفقة المعدلة.
- o خطوة 14: يعتقد المستخدم أن الصفقة الأصلية وردت من قبل الملقم دون أي اعتراض.



الهجوم على المضيف الهدف "Client side Attacks"

في الهجوم على المضيف، يحاول المهاجم استغلال نقاط الضعف الموجودة في تطبيقات المضيف عن طريق إجبارهم على التفاعل مع ملقم خبيث أو عن طريق إجبار التطبيقات معالجة بيانات خبيثة. هناك فرصة كبيره لظهور هذا النوع من الهجوم وهو عندما يتفاعل العميل مع الملقم. إذا لم يتفاعل العميل، فان البيانات الخبيثة لا يمكن أن يتم ارسالها من الخادم. وهكذا، تكون تطبيقات العميل آمنة. أحد الأمثلة على ذلك هو تشغيل بروتوكول نقل الملفات **FTP** العملاء دون اتصال إلى الخادم. عندما لا يوجد أي تفاعل بين العميل والخادم، فان عميل **FTP** يكون في مأمن من هذا النوع من الهجوم.

مثال على أحد التطبيقات التي هي عرضة للهجوم من جانب العميل هو تطبيق الرسائل الفورية "***instant messaging application***". عندما يبدأ هذا التطبيق، فانه يتم اعداد العميل لتسجيل الدخول إلى ملقم بعيد. ويمكن إجراء هجمات ***Client-side attacks*** من خلال ثلاث طرق:

XSS: هجمات **Cross-Site scripting** وهي نوع من هجمات الحقن، التي يتم فيها حقن البرامج النصية الخبيثة في المواقع.

Malicious JavaScript Codes: المهاجم قد يقوم بتضمين جافا سكريبت خبيث في صفحة الويب والذى يقوم بإغرائك لزيارة تلك الصفحة. عند فتح تلك الصفحة في المتصفح الخاص بك، فان الاسكريبت الخبيث يعمل بصمت دون عرض أي رسالة تحذير.

Trojans: حصان طروادة هو تطبيق خبيث التي يتظاهر بأنه مشروع ولكن الغرض الحقيقي هو السماح للقراصنة الوصول الغير مصرح به إلى جهاز الكمبيوتر.



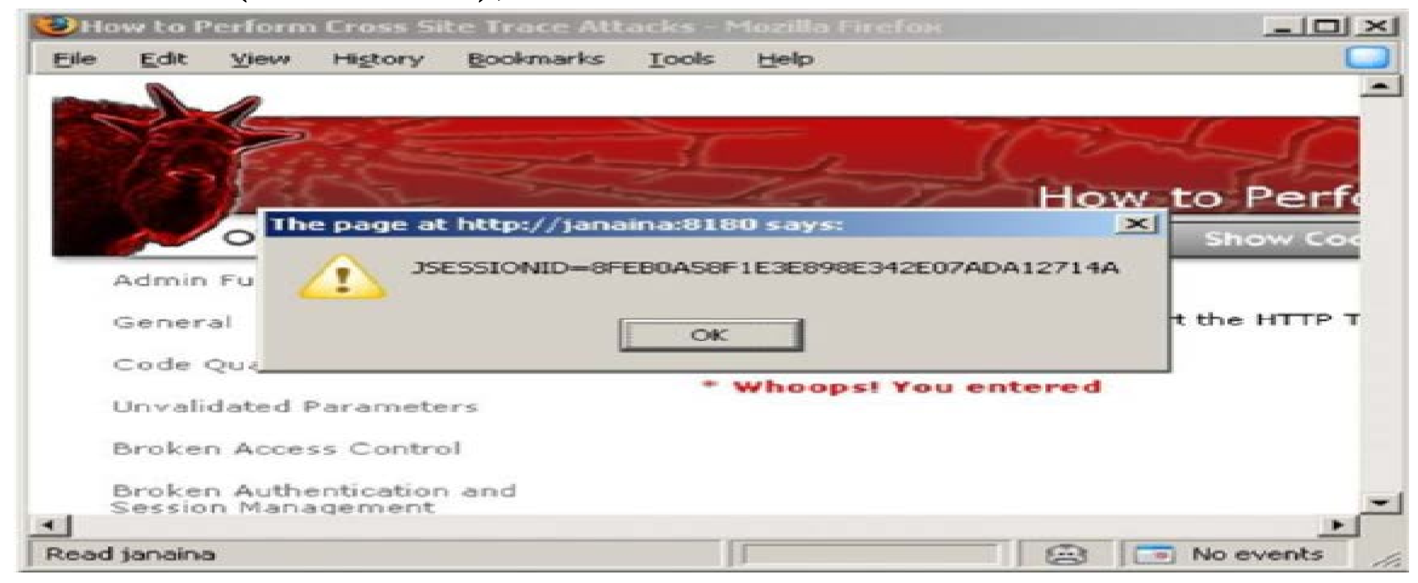
Cross-site Script Attacks

Cross-site scripting هو نوع من نقاط الضعف في أمن الكمبيوتر. وعادة ما وجدت هذه الثغرة الأمنية في تطبيقات الويب حيث يوجد نطاق حقن نص من جانب العميل في صفحات الويب. هذا الضعف يمكن أن يستخدم لتجاوز ضوابط الوصول. المهاجم يحقن الاسكريبت الخبيث من جانب العميل في صفحات الويب ويرسلها الى الضحية المستهدفة لأداء هجوم **cross-site script attack**.

Cross-site script attack هو هجوم من جانب العميل حيث يقوم المهاجم باختراق رمز الجلسة "**session token**" وذلك من خلال الاستفادة من الأكواد أو البرامج الخبيثة. وسوف يذكر مثال هنا لإظهار كيف يقوم المهاجم بسرقة رمز الجلسة "**session token**" وذلك باستخدام هجوم **XSS**. المهاجم يقوم أولا بإرسال وصلة وضعت للضحية مع جافا سكريبت خبيث. ينتظر المهاجم الضحية أو المستخدم للنقر على الرابط. بمجرد قيام الضحية بالنقر على الرابط، سيتم تشغيل الجافا سكريبت تلقائيا وينفذ التعليمات التي قدمها المهاجم. في

هذا المثال يستخدم المهاجم الهجوم **XSS** لعرض قيمة **Cookies** للجلسة الحالية. باستخدام نفس التقنية، فمن الممكن إنشاء شفرة جافا سكريبت محددة التي سوف ترسل ملف **cookies** الى المهاجم.

<SCRIPT>alert (document.cookie) ;</SCRIPT>



Session Fixation

Session fixation هو هجوم يتم أجراءه لاختطاف جلسة عمل لمستخدم صالحة. لتنفيذ هذا الهجوم، فان المهاجم يستفاد من الحد الموجود في إدارة معرفات الجلسة لتطبيق الويب "*web application session ID management*". تطبيق ويب يسمح للمستخدم بالمصادقة باستخدام معرف جلسة قائمة بدلا من توليد معرف جلسة عمل جديدة. في هذا الهجوم، يوفر المهاجم معرف جلسة لتطبيق ويب شرعي ومن ثم يجذب الضحية لاستخدامها. إذا كان متصفح الويب للضحية يستخدم نفس معرف الجلسة، فان المهاجم يمكنه اختطاف جلسة المستخدم الصالحة حيث ان المهاجم على بينة من معرف الجلسة المستخدمة من قبل الضحية.

هجوم **session fixation attack** هو نوع من هجوم اختطاف الجلسة. الفرق بين الهجومين هو أنه، في اختطاف الجلسة يتم تنفيذ الهجوم عن طريق سرقة الجلسة التي أنشئت بعد تسجيل دخول المستخدم في حين أنه في **session fixation attack**، يبدأ الهجوم قبل تسجيل دخول المستخدم وهذا الهجوم يمكن أن يؤدي باستخدام تقنيات مختلفة. هذه التقنية التي يحتاجها المهاجم لاختيار الهجوم يعتمد على كيفية تعامل تطبيق ويب مع رموز الجلسة. وفيما يلي الأساليب الأكثر شيوعا لهجوم **session fixation attack:**

- Session token in the URL argument
- Session token in a hidden form field
- Session ID in a cookie

في طريقة استجابة **HTTP header** في هجمات **session fixation attacks**، المهاجم يستكشف استجابات الملقم لإصلاح **ID** الجلسة. المهاجم قادر على إدراج قيمة معرف الجلسة في ملف **cookie** بمساعدة **Set-Cookie parameter**. بمجرد ان يتم تعيين ملف **cookie**، فان المهاجم يرسله إلى متصفح الضحية.

يتم session fixation attacks على ثلاث مراحل:

Session set-up phase: في هذه المرحلة، المهاجم يحصل لأول مرة على معرف جلسة شرعي بجعل اتصال مع التطبيق على شبكة الإنترنت. قليل من تطبيقات الويب تدعم ميزة **idle session time-out**. في مثل هذه الحالات، المهاجم يحتاج لإرسال الطلبات مرارا وتكرارا من اجل الحفاظ على معرف الجلسة على قيد الحياة.

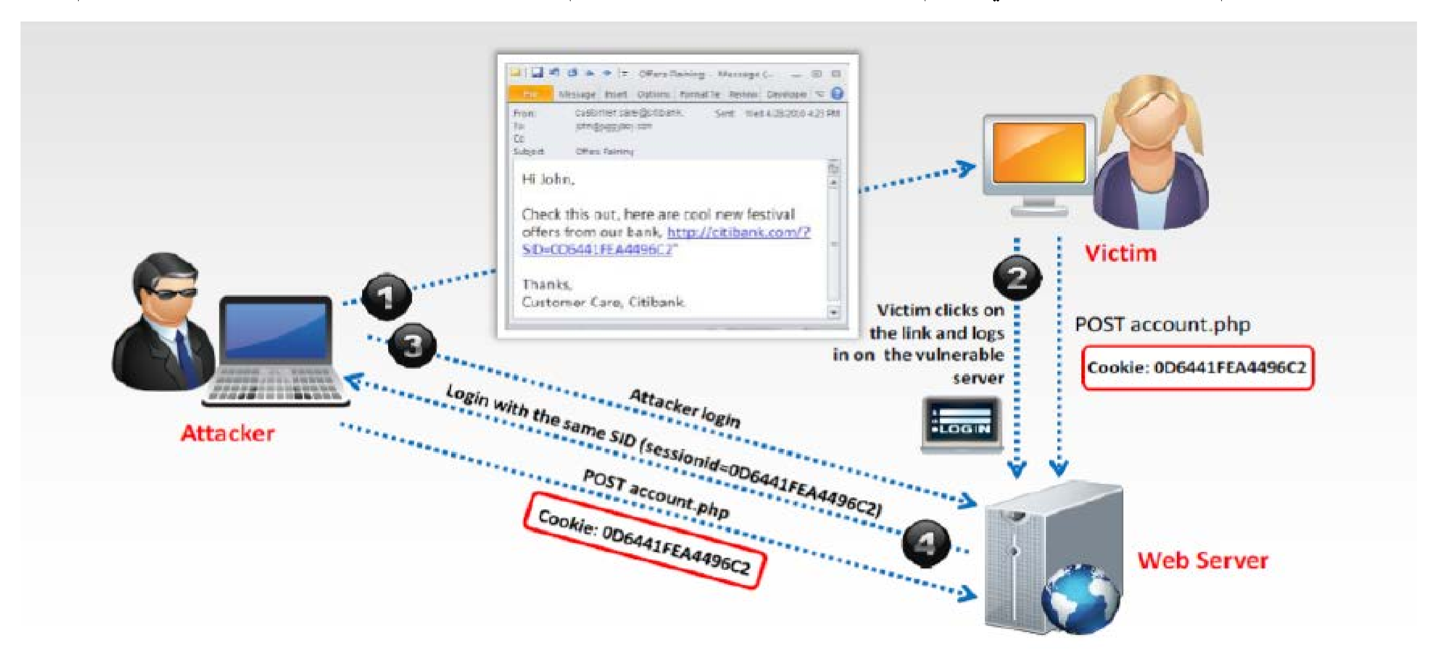
Fixation phase: في هذه المرحلة، المهاجم يقوم بإدراج معرف جلسة لمتصفح الضحية ومن ثم إصلاح الجلسة.

Entrance phase: في هذه المرحلة، المهاجم ينتظر قيام الضحية بتسجيل الدخول إلى خادم الويب باستخدام معرف الجلسة الفخ.

نفترض بأن الضحية يريد استخدام الخدمات المصرفية عبر الإنترنت. دعونا ننظر الى بنك على الانترنت، وليكن مثلا **citibank.com** إذا كان المهاجم يريد استخدام هذه الجلسة، فانه يحتاج إلى اتباع الخطوات المذكورة على النحو التالي:

- أولا، يجب أن يقوم المهاجم بتسجيل الدخول إلى موقع البنك على الانترنت كمستخدم موثوق به.
- ثم يقوم http://citibank.com بإصدار معرف جلسة عمل، وليكن مثلا **0D6441FEA4496C2** الى المهاجم.
- المهاجم يقوم بإرسال الرابط الخبيث الذي يحتوي على معرف جلسة، وليكن **http://citibank.com/? SID=0D6441FEA4496C2**، الى الضحية وخداع الضحية للنقر على ذلك.
- عندما ينقر الضحية على الرابط يتعامل معها باعتبارها رابط شرعي أرسل من قبل البنك، فإنه يوجه الضحية إلى خادم الويب للبنك مع معرف الجلسة **SID=0D6441FEA4496C2**.
- خادم الويب يفحص ويعلم أن معرف الجلسة **0D6441FEA4496C2** أنشئت بالفعل وهو في حالة نشطة، وبالتالي ليس هناك حاجة لإنشاء جلسة جديدة. هنا، يدخل الضحية اسم المستخدم وكلمة المرور للدخول والوصول إلى حسابه.
- الآن يمكن للمهاجم أيضا الوصول إلى جلسة المستخدم الصالحة، أي صفحة الحساب المصرفي للضحية عبر الإنترنت باستخدام **http://citibank.com/? SID=0D6441FEA4496C2** حيث أن المهاجم لديه معرفة بمعرف الجلسة المستخدمة من قبل الضحية.

لتلخيص هذا الهجوم، يمكننا أن نقول إن في هجوم **session fixation attack**، يتم جذب الضحية لتسجيل الدخول إلى جلسة المهاجم.



11.3 اختطاف الجلسة على مستوى الشبكة (Network Level Session Hijacking)

حتى الآن، قد ناقشنا مفاهيم اختطاف الجلسة المختلفة واختطاف الجلسة على مستوى التطبيق. الآن سوف نناقش اختطاف الجلسة على مستوى الشبكة. هذا القسم يسلط الضوء على خطف الجلسة مستوى الشبكة ومختلف التقنيات المستخدمة لأداء هذا النوع من الهجوم.

يتم تنفيذ الاختطاف على مستوى الشبكة "**Network-level hijacking**" على تدفق البيانات من بروتوكول مشترك من قبل جميع تطبيقات الويب. الهجمات على **network-level sessions** توفر للمهاجم المعلومات الهامة التي هي مفيدة في مهاجمة الجلسات على مستوى التطبيق.

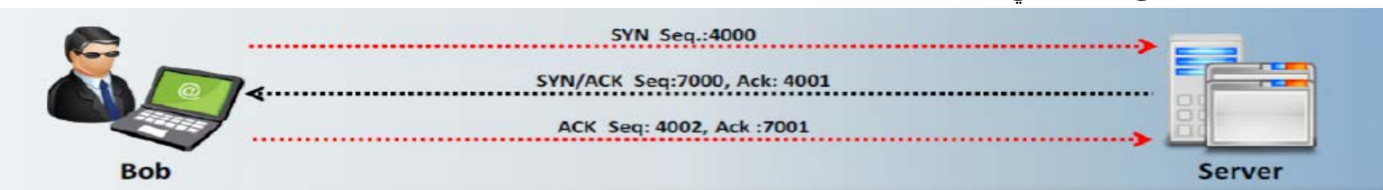
Network-level hijacking تشمل ما يلي:

- TCP/IP hijacking
- IP spoofing: source routed packets
- RST hijacking
- Blind hijacking
- Man-in-the-middle: packet sniffer
- UDP hijacking



The Three-way Handshake

عندما تضع اتصال بين طرفين باستخدام TCP، وإجراء المصافحة الثلاثية. المصافحة الثلاثية "The Three-way Handshake" تبدأ الاتصال وتبادل جميع المعلمات اللازمة لاتصال الطرفين. يستخدم TCP المصافحة الثلاثية لتأسيس اتصال جديد. يوضح الشكل التالي كيف يعمل هذا التبادل هو على النحو التالي:



في البداية، يكون الاتصال على جانب العميل في حالة مغلقة وعلى جانب الملقم في حالة استماع. يبدأ العميل بالاتصال عن طريق إرسال رقم تسلسل الأولي (ISN) ووضع SYN flag. الآن حالة العميل تكون في SYN-SENT state.

عند تلقي هذه الحزمة، فان الملقم يقر رقم تسلسل العميل، ويرسل ISN الخاص به مع اعداد SYN flag. الان تكون حالة الملقم SYN-RECIVE. عند تلقي هذه الحزمة، يقر العميل رقم تسلسل الخادم عن طريق زيادة ذلك ووضع ACK Flag.

العميل الآن في وضع established state. عند هذه النقطة، تم أنشأت جلسة بين الجهازين، ويمكن أن يبدأ التواصل.

عند تلقي Ack العميل، يدخل الخادم في حالة established state ويرسل إلى الخلف acknowledgment، مع تزايد رقم التسلسل العميل. ويمكن إغلاق الاتصال من قبل أي منهما باستخدام المعلم FIN أو RST أو انتهاء المهلة.

إذا تم تعيين إشارة RST مع الحزمة، فان المضيف المتلقي يدخل في حالة مغلقة "CLOSED State" ويحرر جميع الموارد المرتبطة بهذا الاتصال. يتم إسقاط أي من الحزم الواردة الإضافية لهذا الاتصال.

إذا تم إرسال الحزمة مع FIN flag، المضيف المتلقي يقوم بإغلاق الاتصال لأنه يدخل في الوضع CLOSE-WAIT mode. يتم قبول الحزم المرسلة من قبل العميل في اتصال القائم إذا كان رقم التسلسل ضمن النطاق ويتبع سابقتها.

إذا كان عدد التسلسل هو خارج نطاق أرقام التسلسل المقبولة، يتم إسقاط الحزمة، وسوف يتم إرسال حزمة ACK باستخدام رقم التسلسل المتوقع.

الأطراف الثلاثة للتواصل، والأشياء المطلوبة هي كما يلي:

- عنوانIP.
- أرقام المنافذ.
- أرقام التسلسل.

معرفة عنوان IP ورقم المنفذ سهلة؛ حيث انها يتم سردها في حزم IP، والتي لا تتغير على طوال الجلسة. بعد اكتشاف العناوين التي تتم التواصل مع المنافذ، فانه يتم تبادل المعلومات ويبقى على حاله للفترة المتبقية من الجلسة. ومع ذلك، تتغير أرقام التسلسل. ولذلك، يجب على المهاجم بنجاح تخمين أرقام التسلسل لخطف الجلسة "blind hijack". إذا كان المهاجم يمكنه أن يخدع الملقم في تلقي الحزم المغشوشة وتنفيذها، فانه يكون قد نجح في خطف الجلسة.

على سبيل المثال:

- بوب يبدأ اتصال مع الخادم عن طريق إرسال حزمة إلى الملقم مع مجموعة بت SYN.
- يستلم الملقم هذه الحزمة والردود عن طريق إرسال حزمة مع SYN/ACK و ISN(الأولي عدد تسلسل) للملقم.
- بوب بتعيين بت ACK أي أنه يعترف باستلام الحزمة وزيادة رقم التسلسل بمقدار 1.
- قد أنشأ الجهازين جلسة اتصال بينهما بنجاح.

Sequence Numbers

لقد تم بالفعل مناقشة المصافحة الثلاثية في **TCP**. يوفر **TCP** اتصال موثوق ثنائية الاتجاه "**full-duplex**" بين اثنين من النهايات. يتم تعريف الاتصال الفريد من أربعة عناصر: عنوان **IP** المرسل، رقم منفذ **TCP** المرسل وعنوان **IP** جهاز الاستقبال، ورقم منفذ **TCP** المتلقي. يمكن أن ينظر إلى تزايد أعداد رقم التسلسل في المصافحة الثلاثية. كل بايت أرسل من قبل المرسل يحمل رقم تسلسل معين الذي أقر من قبل المتلقي في نهايتها. المتلقي يستجيب إلى المرسل مع رقم التسلسل نفسه. لأغراض أمنية، رقم التسلسل مختلف في الاتصالات المختلفة، ولكل جلسة في اتصال **TCP** لديه رقم تسلسل مختلف. أرقام التسلسل هذه حاسمة للأمن: هم 32 بت، لذلك هناك أكثر من 4 مليارات من التوليفات الممكنة، مما يجعل من الصعب جدا التكهن بها. كما أنها حاسمة للمهاجم في اختطاف الجلسة.

ماذا يحدث عندما يكون رقم التسلسل الأولي (الحزم الأولى من حزمة **SYN** العميل أو حزمة الملقم **(SYN/ACK)** يمكن التنبؤ به؟ عندما يمكن التنبؤ برقم تسلسل **TCP**، فان المهاجم يمكنه إرسال الحزم المزورة لتبدو وكأنها صادرة من جهاز كمبيوتر موثوق به. يمكن للمهاجمين أيضا تنفيذ اختطاف الجلسة للوصول إلى معلومات غير مصرح بها.

الخطوة التالية هي لتشديد تنفيذ نظام التشغيل من **TCP** وإدخال العشوائية في **ISN**. ويتم ذلك من خلال استخدام عدد من مولدات المزيف **(PRNGs)**. وعشوائية **ISNS** المستخدمة في اتصالات **TCP** باستخدام **PRNGs**. ومع ذلك، بسبب الآثار المترتبة على نظرية النهاية المركزية، اضافة سلسلة من الأرقام توفر اختلاف غير كافي في نطاق قيم **ISN** المحتملة، مما يسمح للمهاجمين تعطيل أو خطف اتصالات **TCP** قائمة أو تزوير اتصالات مستقبلية ضد **vulnerable TCP/IP stack implementations**. وهذا يعني أن الأنظمة تعتمد زيادات عشوائية لتوليد **ISNS** والتي لا تزال عرضة للهجوم الإحصائي. بعبارة أخرى، مع مرور الوقت، وحتى أجهزة الكمبيوتر ذات اختيار لأرقام عشوائية يكررون أنفسهم لأنه يعتمد على العشوائية على خوارزمية داخلية التي يستخدمها نظام تشغيل معين. بمجرد الاتفاق على رقم التسلسل، فان كل الحزم التي تتبع ستكون **ISN_1**. وهذا يجعل ضخ البيانات في مجرى الاتصالات ممكن.

وفيما يلي بعض المصطلحات المستخدمة في اشارة الى أرقام ISN:

- **SVR_SEQ**: رقم تسلسل البايت التالي ليتم إرسالها من قبل الملقم.
- **SVR_ACK**: البايت التالي التي يتم استلامها من قبل الخادم (رقم تسلسل البايت الأخير الذى تلقى زائد واحد).
- **SVR_WIND**: اطار الملقم المتلقي.
- **CLT_SEQ**: رقم تسلسل البايت التالي ليتم إرسالها من قبل العميل.
- **CLT_ACK**: البايت التالي ليتم استلامها من قبل العميل.
- **CLT_WIND**: اطار العميل المتلقي.

في البداية، لا يتم تبادل أية من البيانات، وهذا هو، **CLT_ACK _ SVR_SEQ**و**SVR_ACK CLT_SEQ**. هذه المعادلات صحيحة أيضا عندما يكون الاتصال في حالة هادئة "**quite state**"، هذا هو، لا يتم إرسال أية من البيانات على كل جانب. هذه المعادلات ليست صحيحة أثناء حالات عابرة عندما يتم إرسال البيانات. وفيما يلي مجالات حزمة **TCP header**:

- **Source port**: رقم منفذ المصدر.
- **Destination port**: رقم منفذ الوجهة.
- **Sequence number**: رقم التسلسل من البايت الأول في هذه الحزمة.
- **Acknowledgment number**: رقم التسلسل المتوقعة من البايت المقبل.

وفيما يلي بت التحكم:

URG: Urgent pointer
ACK: Acknowledgment
PSH: Push function
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: No more data from sender
Window: Window size of the sender
Checksum: TCP checksum of the header and data
Urgent pointer: TCP urgent pointer
Options: TCP options
SEG_SEQ: Refers to the packet sequence number (as seen in the header)

SEG_ACK: Refers to the packet acknowledgment number
SEG_FLAG: Refers to the control bits

في حزمة نموذجية يتم إرسالها من قبل العميل (لا يوجد أي إعادة إرسال)، يتم تعيين **SEG_SEQ** إلى **CLT_SEQ**، و **SEG_ACK** إلى

CLT_ACK. CLT_ACK <_ SVR_SEQ <_ CLT_ACK _ CLT_WIND SVR_ACK <_CLT_SEQ <_SVR_ACK _ SVR_WIND.

إذا كان العميل يبدأ الاتصال مع الخادم، فإن الإجراءات التالية تتم:

- الاتصال على جانب العميل هو في حالة مغلقة.
- الاتصال على جانب الملقم في حالة الاستماع.
- يرسل العميل أولا رقمه تسلسل الأولي ويحدد **SYN bit** الى **SEG_SEQ = CLT_SEQ_O, SEG_FLAG = SYN**.
- حالته الآن هي **SYN-SENT**.
- عندما يستلم الملقم هذه الحزمة، فإنه **Ack** رقم تسلسل العميل، ويرسل **ISN** الخاص به، ويحدد بت **SYN** الى:

SEG_SEQ_ SVR_SEQO
SEQACK _ CLT_SEQ_0_1
SEG_FLAG _ SY N
And sets:
SVR_ACK_CLT_SEO_O_1

الآن **SYN** في حالة الاستقبال "**SYN-RECEIVED**".

- عند تلقي هذه الحزمة، فان العميل يقر **ISN** الخادم:

SEG_SEQ _ CLT_SEQO_1
SEG_ACK _ SVR_SEO_0_1
And sets CLT_ACK_SVR_SECLO_1

- حالته الان **ESTABLISHED**.
- على تلقي هذه الحزمة يدخل الخادم حالة **ESTABLISHED**.

CLT_SEQCLT_SECLO_1
CLT_ACK_SVR_SEQO_1
SVR_SEQSVR_SEQO_1
SVR_ACK_CLT_SEQ_O_1

- يظهر النص الخطوات التالية في العملية.

Server	Client
LISTEN	CLOSED
	<-SYN,
	CLT_SEQ_0
LISTEN	SYN_SENT
SYN,ACK->	
SVR_SEQ_0	
CLT_SEQ_0+1	
SYN_RECEIVED	ESTABLISHED
	SVR_SEQ = CLT_SEQ_0+1
	CLT_ACK = SVR_SEQ_0+1
	<-ACK,
	CLT_SEQ_0+1
	SVR_SEQ_0+1
ESTABLISHED	
SVR_SEQ = SVR_SEQ_0+1	
SVR_ACK = CLT_SEQ_0+1	

إذا كان رقم تسلسل ضمن إطار التلقي معروف، فان المهاجم يمكنه ضخ البيانات في مجرى الجلسة أو إنهاء العلاقة إذا كان يعرف عدد البايتات التي تنتقل حتى الآن في الجلسة (لا ينطبق إلا على **blind hijack**).
يمكن للمهاجم تخمين مجموعة مناسبة من أرقام التسلسل ويرسل عددا من الحزم في الشبكة مع أرقام تسلسل مختلفة التي تقع ضمن النطاق المناسب. أذكر أنه يتم استخدام حزمة **FIN** لإغلاق الاتصال. بمجرد معرفة النطاق، فمن المرجح أن الملقم يقبل حزمة واحدة على الأقل. بهذه الطريقة، فان المهاجم لا يرسل حزمة لكل رقم تسلسل، ولكن يمكن اللجوء إلى إرسال عدد مناسب من الحزم مع أرقام تسلسل حجم الإطار على حدة.
ولكن كيف يمكن للمهاجم ان يعرف عدد الحزم ليتم إرسالها؟ يتم الحصول على هذه النسبة بقسمة مجموعة أرقام التسلسل المراد تغطيتها من قبل جزء من حجم الإطار حيث يستخدم كمؤشر الزيادة. **PRNG** يعتني بهذا التوزيع العشوائي. صعوبة تنفيذ مثل هذه الهجمات يتناسب طرديا مع عشوائية **ISNS**. كلما كان **ISN** أكثر عشوائية، كلما كان أكثر صعوبة على المهاجم.

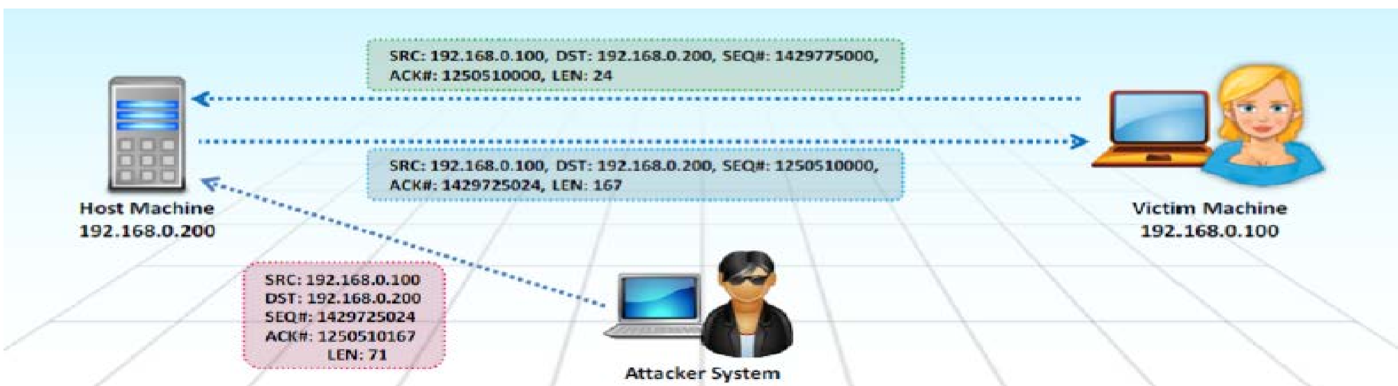
التنبؤ بأرقام التسلسل "Sequence Numbers Prediction"

بمجرد ان يقوم العميل بإرسال طلب الاتصال (**SYN**) في حزمة إلى الملقم، فان الملقم يستجيب (**SYN / ACK**) مع رقم تسلسل، والتي يجب ان يقرها العميل (**ACK**).
رقم التسلسل هذا يمكن التنبؤ به. المهاجم يرتبط بالخدمة أولا مع عنوان **IP** الخاص به، ويسجل رقم التسلسل المختار، ثم يفتح اتصال ثاني مع عنوان **IP** مزورة. المهاجم لا يرى **SYN/ACK** (أو أي حزمة أخرى) من الخادم، ولكن يمكن تخمين الإجابة الصحيحة. إذا تم استخدام عنوان **IP** المصدر للمصادقة، يمكن للمهاجم استخدام الاتصالات من جانب واحد لاقتحام الخادم.

TCP/IP Hijacking

TCP/IP hijacking هو أسلوب قرصنة التي تستخدم الحزم المنتحلة للاستلاء على اتصال بين الضحية والجهاز المضيف. النظم التي تستخدم كلمات السر لمرة واحدة يمكن مهاجمتها بسهولة من خلال هذه التقنية. اتصال الضحية يصبح معلق والمهاجم قادر على التواصل مع الجهاز المضيف كما لو كان المهاجم هو الضحية. ويمكن أن يؤديها على النظام على نفس الشبكة مثل الضحية. الجهاز المضيف يمكن أن يكون موجودا في أي مكان.
الخطوات التي يتعين القيام بها في اختطاف **TCP/IP hijacking** هي كالاتي:

- التنصت على اتصال الضحية من خلال حصوله على أرقام التسلسل له.
- استخدام رقم التسلسل، المهاجم يرسل حزمة منتحلة من نظام الضحية إلى النظام المضيف.
- الجهاز المضيف يستجيب للضحية، على افتراض أن الحزمة وصلت منه، وبالتالي يزيد عدد التسلسل وبالتالي الاستجابة لـ **IP** الضحية.



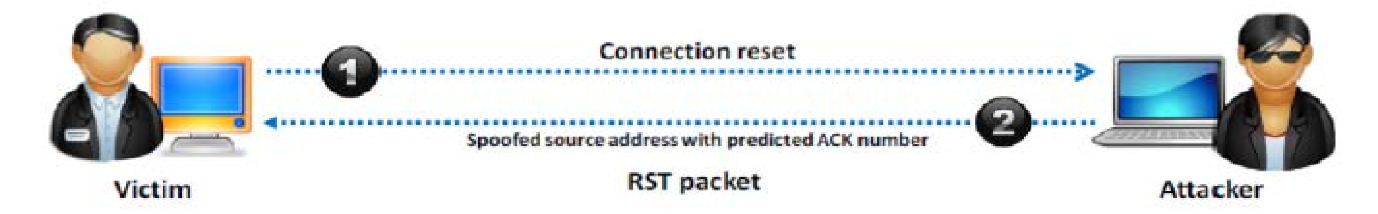
TCP/IP hijacking هي تقنية خطيرة يستخدمها المهاجمون للوصول إلى المضيف في شبكة ومن ثم فصله عن الشبكة منطقيا. للوصول إلى المضيف، المهاجم يتنصت في البداية على اتصال الضحية ويستخدم **IP** الضحية لإرسال حزمة منتحلة مع رقم التسلسل المتوقع. المضيف يعالج الحزمة المنتحلة، بزيادة الرقم المتسلسل، ويرسل الإقرار إلى **IP** الضحية. آلة الضحية تجهل الحزمة المنتحلة، لذلك يتجاهل **ACK** حزمة الجهاز المضيف وتتحول رقم تسلسل العد قبالة. لذلك، يتلقى المضيف الحزم مع رقم تسلسل غير صحيح. يجبر المهاجم اتصال الضحية مع الجهاز المضيف إلى حالة غير متزامنة. المهاجم يتعقب أرقام التسلسل وبشكل مستمر يزيف الحزم التي تأتي من **IP** الضحية. يواصل المهاجم التواصل مع الجهاز المضيف بينما يعلق اتصال الضحية.

IP Spoofing: Source Routed Packets

يتم استخدام تقنية IP المزيفة "IP Spoofing" من أجل الوصول غير المصرح به إلى أجهزة الكمبيوتر. المهاجم يرسل رسالة إلى الخادم مع عنوان IP يشير إلى أن الرسالة من مجموعة موثوق بها. أولا، المهاجم يحصل على عنوان IP من العميل وتعديل رؤوس الحزم للإشارة إلى أنه يأتي من عنوان IP موثوق به. هذا النوع من الخطف يسمح للمهاجمين لخلق الحزم المقبولة الخاصة لتضاف الى جلسة TCP.
الحزم يتم توجيهها الى المصدر، حيث يحدد المرسل مسار الحزم من المصدر إلى IP الوجهة. باستخدام هذه التقنية توجيه المصدر، يمكن المهاجمين خداع الخادم بالاعتقاد أنه هو يتواصل مع المستخدم.
بعد استخدام عنوان IP المزيف بنجاح، فان المهاجم يغير رقم التسلسل ورقم Ack الذي يتوقعه الخادم. بعد تغيير هذا الرقم، المهاجم يحقن الحزم المزورة في جلسة TCP قبل ان يستجيب العميل لها. وهذا يؤدي إلى حالة الغير متزامن لأن رقم التسلسل وACK غير متزامن بين العميل والخادم.

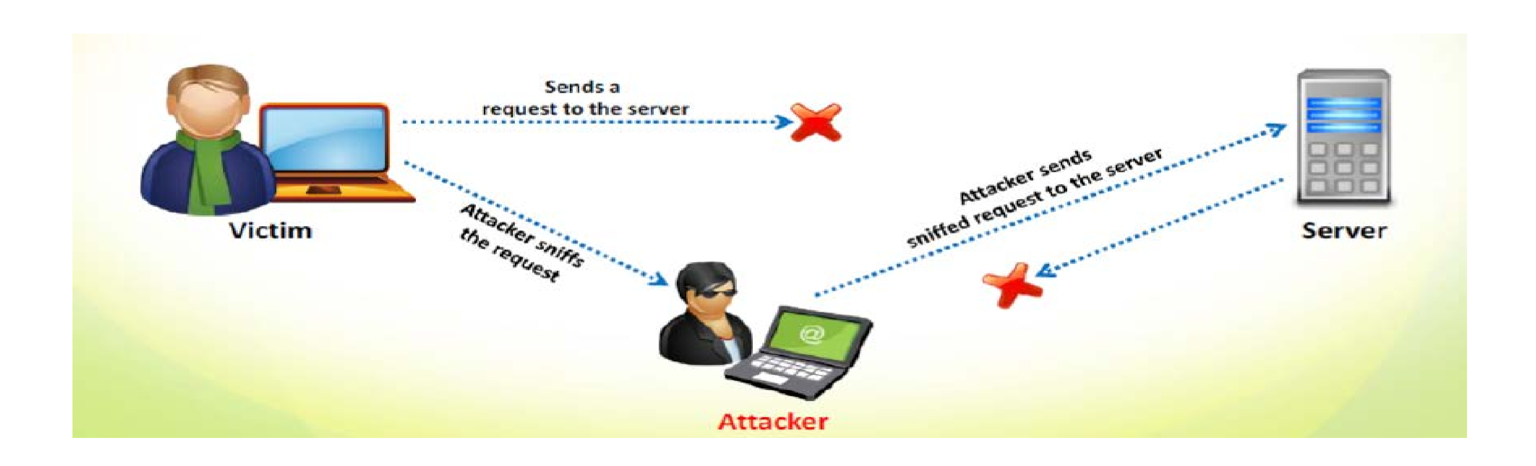
RST Hijacking

RST hijacking هو شكل من أشكال **TCP/IP hijacking** حيث يتم حقن حزم (RST). في هذا الهجوم، المهاجم يتنصت أولا على الاتصال بين المصدر والضحية للاستيلاء على معلومات إنشاء الاتصال مثل عناوين IP المصدر والضحية، أرقام تسلسل، وما إلى ذلك.
المهاجم الآن ينشأ حزمة RST مع عنوان المغشوش وكما انه عنوان المصدر ورقم Ack نفسه كما انه في اتصال حقيقي ومن ثم يرسله إلى الضحية. عندما يتلقى الضحية الحزمة المنتحلة فأنه يعتقد أن طلب rest تم إرسالها من قبل المصدر، وبالتالي يقوم بإعادة تعيين الاتصال.
يمكن أن يتم **RST hijacking** باستخدام أداة لصياغة الحزم مثل **Colasoft's Packet Builder**. يمكن لأدوات أخرى مثل **TCPDUMP**، **AWK**، و**nemesis** المساعده في إعادة الاتصال. **TCPDUMP** يمكنه الكشف عن الاتصالات التي تم تأسيسها من خلال فلرتة الحزم التي تحتوى على **ACK flag**. **AWK** هو أداة تقوم بتوزيع الناتج التي حصلنا عليه من الأداة **TCPDUMP** لاشتقاق عناوين المصدر والواجهة، والمنافذ، وعناوين **MAC**، ورقم التسلسل، وأرقام **ack**. **RST hijacking** هو نوع من هجوم حجب الخدمة حيث يتم رفض الوصول إلى الخدمة أو الموارد.



Blind Hijacking

Blind hijacking ينطوي على توقع أرقام التسلسل التي يرسلها المضيف الضحية من أجل إنشاء اتصال يبدو أنه صادر من المضيف. قبل الاستكشاف بالتحايل الأعمى "blind spoofing"، فلنلقي نظرة على التنبؤ برقم التسلسل. أرقام تسلسل TCP، والتي هي فريدة من نوعها لكل بايت في جلسة TCP، توفر التحكم في التدفق وسلامة البيانات نفسها. وبالإضافة إلى ذلك، TCP segment يعطي رقم تسلسل أولي (ISN) كجزء من segment header. لا يبدأ رقم التسلسل الأولي عند مستوى الصفر لكل جلسة. Participants' state ISNs كجزء من عملية المصافحة في اتجاهات مختلفة، والبايت يتم ترقيمهما بالتتابع. يعتمد Blind IP hijacking على قدرة المهاجم على التنبؤ بأرقام التسلسل، لأنه غير قادر على التنصت على التواصل بين المضيفين الاثنين بحكم انه ليس على نفس جزء الشبكة. المهاجم لا يمكنه تزييف مضيف موثوق على شبكة مختلفة ورؤية حزم الرد لأن الحزم لا توجه له. بالإضافة المهاجم لا يمكنه اللجوء الى ARP cache poisoning بسبب ان الراوتر لا يقوم ببث ARP عبر الإنترنت. كما ان المهاجم غير قادر على رؤية الردود، فان يضطر الى استباق الردود من الضحية ومنع المضيف من إرسال RST للضحية. المهاجم يحقن نفسه في الاتصال من خلال التنبؤ بأرقام تسلسل المضيف البعيد والتي يتوقعها من الضحية.
في **Blind hijacking**، المهاجم يسرد التخمينات بشكل صحيح حول ISN القادم من جهاز كمبيوتر الذي يحاول تأسيس الاتصال. يمكن للمهاجم ان يرسل الأوامر، مثل وضع كلمة مرور للسماح بالوصول الى موقع آخر على الشبكة، ولكن لا يمكنه أبدا رؤية استجابة. يمكن للمهاجم حقن البيانات الخبيثة أو الأوامر في الاتصالات التي تم اعتراضها في جلسة TCP حتى إذا تم تعطيل source-routing.



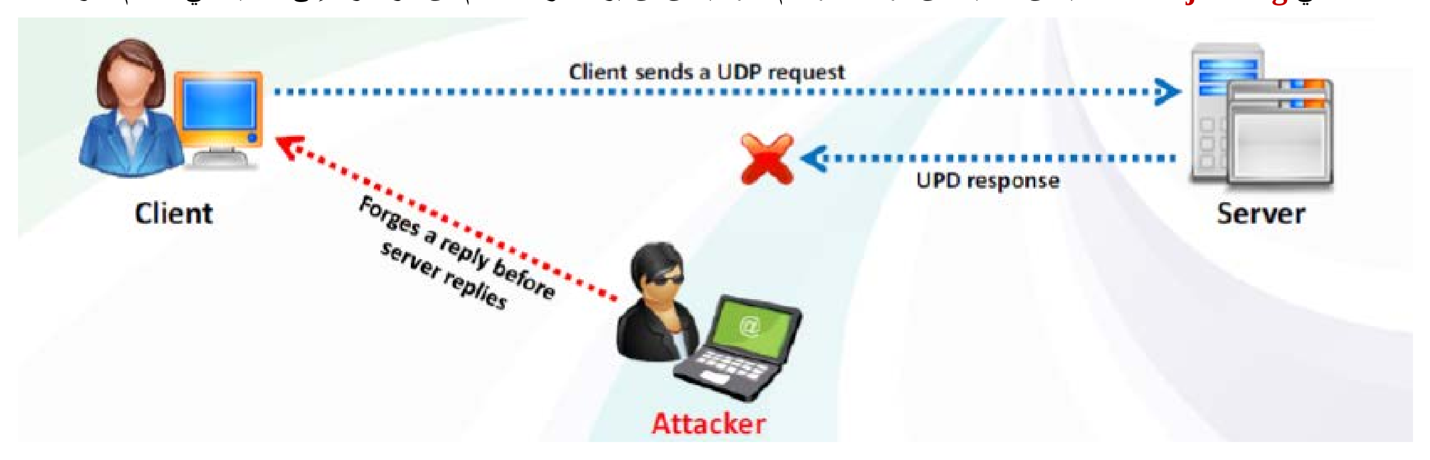
Man-in-the-Middle Attack using Packet Sniffer

Man-in-the-middle يستخدم **packet sniffer** لاعتراض الاتصالات بين العميل والخادم. المهاجم يقوم بتغيير **gateway** الافتراضية لجهاز العميل ويعتزم توجيه الحزم من خلال **hijacker's host**. التقنية المستخدمة هي صياغة حزم **ICMP** لإعادة توجيه حركة المرور بين العميل والمضيف من خلال **hijacker's host**. وتستخدم هذه لإرسال رسائل الخطأ التي تشير إلى مشاكل في معالجة الحزم من خلال الاتصال وخداع الخادم والعميل للتوجيه من خلال مساره.

أسلوب آخر مستخدمة هو تزيف **ARP**. وتستخدم **ARP table** من قبل المضيفين لتعيين عناوين **IP** المحلية إلى عناوين الأجهزة أو عناوين **MAC**. المهاجم يرسل ردود **ARP** مزورة التي تقوم بتحديث جداول **ARP** في المضيف الذي يقوم ببث طلبات **ARP**. وبدلا من ذلك يتم تسليم حركة المرور المرسلة إلى هذا **IP** بدلا من المضيف.

UDP Hijacking

UDP لا تستخدم تسلسل الحزم والمزامنة، لذلك يمكن للمهاجم مهاجمة جلسة **UDP** بسهول عن **TCP**. في هذا الهجوم، **hijacker** يشق رد الخادم لطلب **UDP** العميل قبل أن يستجيب الخادم. الرد على الخادم يمكن أن يقتصر بسهولة إذا تم استخدام التنصت. هجوم رجل في منتصف في **UDP hijacking** يمكن التقليل من مهمة المهاجم لأنها يمكن أن يوقف رد الخادم من الوصول إلى العميل في المقام الأول.



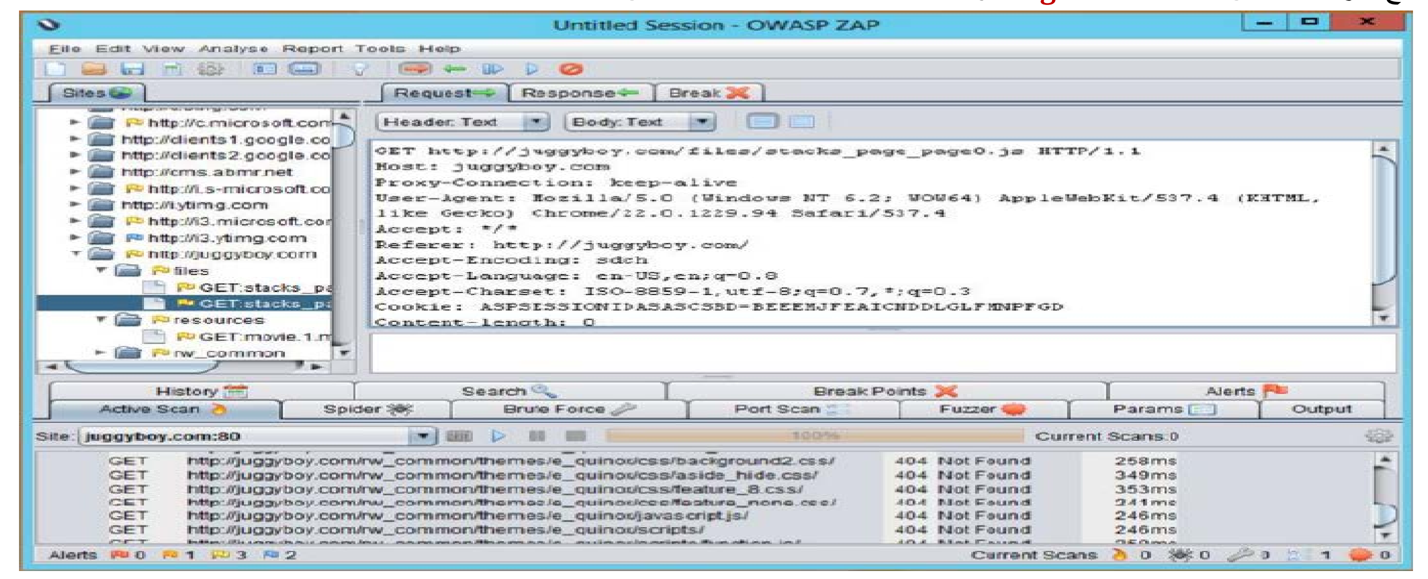
11.4 أدوات اختطاف الجلسة (Session Hijacking Tools)

حتى الآن، قد ناقشنا اختطاف الجلسة ومفاهيمه، على مستوى التطبيق وعلى مستوى الشبكة وتقنيات مختلفة لتنفيذ هجمات اختطاف الجلسة. هذه الأنواع من الهجمات لا يمكن أن يؤديها الا مع مساعدة من الأدوات. أدوات اختطاف الجلسة تجعل وظيفة المهاجم سهله. يسرد هذا القسم وصف مختلف للأدوات المستخدمة من قبل المهاجم لتنفيذ عملية خطف الجلسة.

Session Hijacking Tool: ZAP

المصدر: https://www.owasp.org/index.php/Main_Page

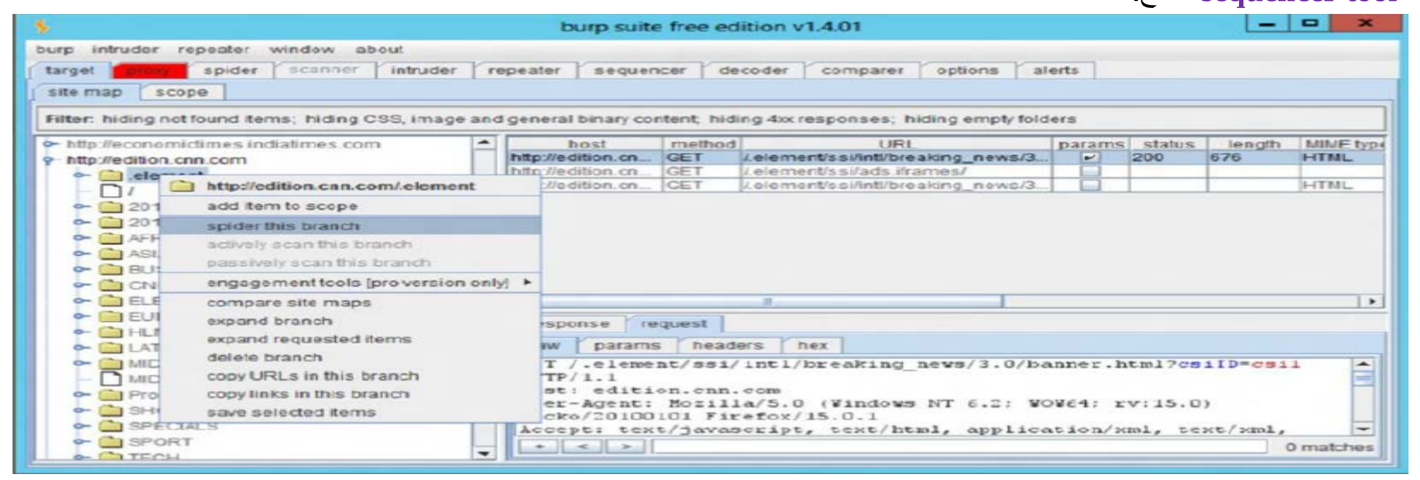
The Zed Attack Proxy (ZAP) هي أداة اختبار الاختراق للعثور على نقاط الضعف في تطبيقات الويب. وهي مصممة لاستخدامها من قبل الاشخاص ذات المجموعة الواسعة من الخبرة الأمنية وعلى هذا النحو يعتبر مثاليا للمطورين ومختبري الوظائف الذين هم جدد على اختبار الاختراق. هذه الأداة لديها الفحص الآلي ومجموعة من الأدوات التي تسمح لك للعثور على الثغرات يدويا. وهو بروكسي اعتراض مع قدرات الفحص **active**، **negative** و**brute force**. لديها كذلك فاحص المنافذ.



Session Hijacking Tool: Burp Suite

المصدر: http://portswigger.net

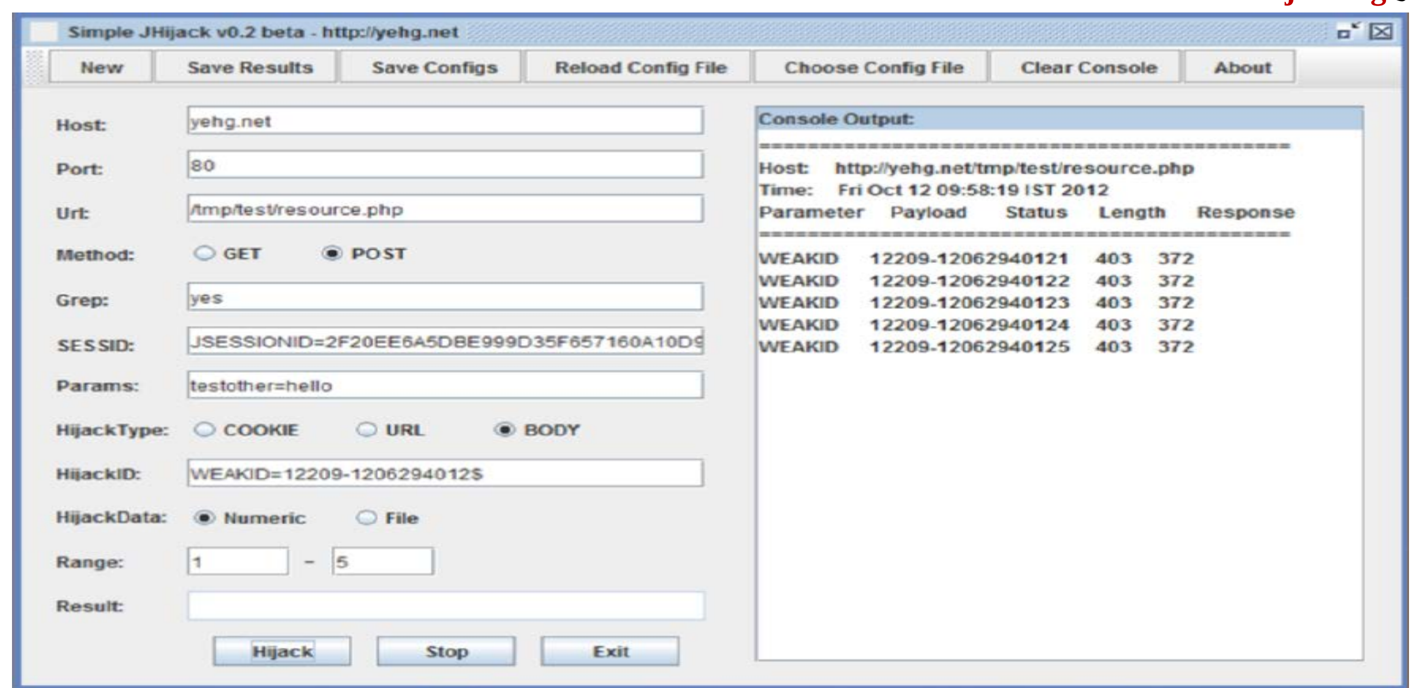
تم تصميم **Burp Suite** خصيصا لاختبار أمن تطبيقات الويب باستخدام برنامجها المتكامل. أدواتها المختلفة تعمل بسلاسة معا لدعم عملية الاختبار بأكملها، من رسم الخرائط الأولى وتحليل سطح الهجوم على التطبيق، وصولا إلى إيجاد واستغلال الثغرات الأمنية. المكونات الرئيسية **Burp Suite** تشمل البروكسي، **spider** الفاحص، أداة الدخيل "**intruder tool**"، أداة مكرر "**repeater tool**"، أداة المنظم "**sequencer tool**"، الخ.



Session Hijacking Tool: JHijack

المصدر: http://jhijack.sourceforge.net

JHijack هي أداة تسمح لك بتقييم الأمن لجلسة تطبيقات الويب. **Java fizzer** تستخدم أساسا لتعداد المعلمة "***parameter enumeration***" و **numeric session hijacking**.



Session Hijacking Tools

بالإضافة إلى **Zaproxy**، **Burp Suite**، و**JHijack**، تتوفر العديد من الأدوات اختطاف الجلسة أخرى. أدوات اختطاف الجلسة هذه تسمح لك لاختطاف جلسة **TCP**. هذه الأدوات حتى تخطف اتصالات **HTTP** لسرقة الكوكيز:

Hamster available at http://erratasec.blogspot.in
Surf Jack available at https://code.google.com
Ettercap available at http://ettercap.sourceforge.net
Hunt available at http://packetstormsecurity.org
TamperlE available at http://www.bayden.com
Ferret available at http://www.erratasec.com
PerJack available at http://packetstormsecurity.org
WhatsUp Gold Engineer's Toolkit available at http://www.whatsupgold.com
Juggernaut available at http://www.securiteam.com
Cookie Cadger available at http://www.cookiecadger.com

11.5 التدابير المضادة (counter measure)

بمجرد إجراء جميع الاختبارات وتحديد نقاط الضعف، بمثابة إنك مختبر الاختراق، فيجب أن تفكر في التدابير المضادة المحتملة التي يمكن أن تحمي الشبكة المستهدفة من القرصنة. يسلط هذا القسم الضوء على مختلف التدابير المضادة ضد اختطاف الجلسة ويسرد أيضا مبادئ توجيهية لمطوري الويب ومجموعة من البروتوكولات التي وضعتها **IETF** لدعم التبادل الآمن للحزم في طبقة **IP**، أي أمن بروتوكول الإنترنت.

الحماية ضد اختطاف الجلسة "Protecting against Session Hijacking"

وفيما يلي طرق الحماية ضد اختطاف الجلسة:

استخدام قذيفة آمنة (SSL) لإنشاء قناة اتصال آمنة: **SSL** هو بروتوكول يستخدم لأمن الاتصالات عبر الإنترنت. و**SSL** يقوم بتشفير شرائح اتصالات الشبكة في طبقة النقل. مع إعداد **SSL** على الشبكة الخاصة بك، يمكنك إرسال أي معلومات سرية مثل أرقام بطاقات الائتمان والعناوين وغيرها من تفاصيل الدفع من خلال شبكة الإنترنت. حتى إذا كان المهاجم يسرق البيانات فانه لا جدوى منها، كما أن **SSL** يخلق اتصال مشفر.

تمرير ملفات الكوكيز الخاصة بعملية المصادقة عبر اتصال HTTPS: **HTTPS** هي النتيجة التي حصل عليها من إضافة قدرات أمنية أو **SSL** للاتصالات **HTTP** القياسية. وعلى غرار **SSL**، **HTTPS** يقدم حماية لملفات الكوكيز.

تنفيذ وظائف الخروج للمستخدم الذي قام بإنهاء الجلسة: واحدة من أهم الخطوات الدفاعية لتجنب اختطاف الجلسة هو تنفيذ وظائف الخروج "**log-out function**". هذا يفرض المصادقة عند بدء جلسة أخرى.

توليد معرف الجلسة بعد الدخول الناجح: هذا يمنع هجمات **session fixation** حيث ان المهاجم لا يكون على علم بمعرف جلسة الذي أنشا بعد تسجيل الدخول.

تمرير البيانات مشفرة بين المستخدمين وخوادم الويب: تشفير البيانات الخاصة بك قبل نقلها عبر شبكة الإنترنت بحيث ان قيام المهاجمين بسرقة البيانات يكونا غير قادرين على فهم الرسالة أو البيانات.

استخدام سلسلة أو رقم عشوائي طويل كمفتاح الجلسة: مفاتيح الجلسة مهمة جدا في مجال الاتصالات. مفتاح الجلسة هذه يمكن تحديده بسهولة مع مساعدة من هجوم **brute forcing**، إذا كان طول الجلسة الرئيسية صغير. وبالتالي، لتجنب هذه المخاطر، يجب عليك استخدام سلسلة أو رقم عشوائي طويل كمفتاح الجلسة.

استخدام أسماء مستخدمين وكلمات مرور مختلفة لحسابات مختلفة: للحصول على الحماية المناسبة للحسابات الخاصة بك على الانترنت يجب عليك أن تستخدم لفترة أطول كلمات السر مع مجموعات مختلفة. كلمات السر أطول تجعل من الصعب على المهاجمين تخمينها أو التلاعب بها. باستخدام أسماء المستخدمين وكلمات المرور المختلفة لحسابات مختلفة يتجنب خطر المساس بجميع الحسابات، عندما ينجح المهاجم في المساومة على حساب واحد.

تقليل الوصول البعيد: تقليل الوصول البعيد يتجنب حقن المهاجمين جلسة الاتصالات للمستخدم الشرعي مع الملقم البعيد.

تثقيف الموظفين: تثقيف الموظفين حول الأنواع المختلفة من هجمات اختطاف الجلسة، علامات، والدفاعات ضد الهجمات. هذا يساعدك على تجنب هجمات اختطاف الجلسة ويساعدك على اتخاذ إجراءات فورية، إذا نجح المهاجم في الخطف.

لا تنقل معرف الجلسة في سلسلة الاستعلام: معرفات الجلسة في سلاسل الاستعلام أو حقول النموذج يمتلك خطر التسريب من خلال المرجعية. ولذلك فمن المستحسن عدم نقل معرفات الجلسات في سلسلة الاستعلام.

الحد من الاتصالات الواردة: هذا يعمل بشكل جيد عندما يكون نطاقات **IP** محدودة ويمكن التنبؤ بها. مثال على مثل هذه البيئة هو الإنترنت.

استخدام switch بدلا من hubs: عادة تنقل البيانات **hubs** إلى جميع الأنظمة المتصلة في الشبكة، الأمر الذي يجعل وظيفة المهاجم سهله. وخلافا لل **hubs**، فان السويتش يرسل البيانات فقط إلى المضيف الوجهة. وبالتالي، تجنب هجمات اختطاف الجلسة، ولذلك يفضل السويتش على **hubs**.

استخدام البروتوكولات المشفرة التي تتوفر في OpenSSH suite: **OpenSSH** هو مجموعة من أدوات اتصال **SSH**. جميع البروتوكولات المشفرة الموجودة حاليا في **OpenSSH** تنقل كلمات السر مشفرة عبر الإنترنت. انه يقوم بترميز أيضا كل حركة المرور ويزيل خطر التنصت، اختطاف الجلسة، وغيرها من الهجمات.

تكوين قواعد التزوير الداخلية والخارجية المناسبة في gateway: لتجنب اختطاف جلسة الشبكة عن بعد (**RNSH**) أو **blind spoofing** فإنك تحتاج إلى تكوين قواعد التزييف الداخلية والخارجية المناسبة على **gateway**.

استخدام منتجات IDS أو ARP watch وذلك لرصد ARP cache poisoning

استخدام مصادقة قوية (مثل كيربيروس) أو peer-to-peer VPNs

يعرف الدفاع في العمق على انه ممارسة استخدام أنظمة أمنية متعددة أو تقنيات لمنع اختراقات الشبكة. بل هو مكون رئيسي من خطة أمنية شاملة وخصوصيه يحمي الشبكة من هجمات خطف الجلسة. الفكرة المركزية وراء هذا المفهوم هو أنه إذا فشل مضاد واحد، فهناك مستويات إضافية من الحماية المتبقية لحماية الشبكة. الدفاع في العمق يبطئ من سرعة المهاجم لتنفيذ هجوم مما يجعل من الضروري بالنسبة له الاختراق من خلال طبقات عديدة من الأمن. وهذا يعطي وقتا إضافيا لمسؤولي الأمن لكشف والدفاع ضد الهجوم.

استراتيجية تكوين جدار الحماية الجديدة هي مثال جيد على استراتيجية الدفاع في العمق. لتحقيق استراتيجية الدفاع في العمق، العديد من شبكات آمنة للغاية تنفذ عدة أنواع من جدار الحماية.

الكشف عن هجمات خطف الجلسة على الشبكات المزدحمة هو مهمة صعبة للغاية. هناك علامات منبهة، مثل قطع اتصال أجهزة الكمبيوتر مع الشبكة أو ازدحام الشبكة، ولكن هذه العلامات عادة تحصل نتيجة تجاهلها من قبل المستخدمين بأنه "مشاكل في الشبكة النموذجية". لحماية الشبكة، فان مسؤولي الشبكة يستغرق عدة خطوات. الدفاع في العمق أمر بالغ الأهمية لإنشاء خطة أمنية فعالة.

Methods to Prevent Session Hijacking: To be Followed by Web Developers

عادة ما يتم إجراء اختطاف الجلسة من خلال استغلال نقاط الضعف في الآليات المستخدمة لإنشاء الجلسة. مطوري الويب غالبا ما تتجاهل الأمن. خلال عملية التطوير، إذا نظر في مطوري الويب الى المبادئ التوجيهية المذكورة التي تليها، فان خطر اختطاف الجلسة يمكن تجنبها إلى حد ما:

- إنشاء مفاتيح جلسة مع سلاسل طويلة أو رقم عشوائي بحيث يكون من الصعب على المهاجم تخمين مفتاح جلسة عمل صالحه.
- تشفير البيانات ومفتاح الجلسة التي يتم نقلها بين المستخدم وخوادم الويب.
- منع التنصت داخل الشبكة.
- تجديد معرف الجلسة بعد الدخول الناجح لمنع هجوم **session fixation attack**.
- تنتهي الجلسة في أقرب وقت من تسجيل المستخدم للخروج.
- تقليل العمر الافتراضي للجلسة أو الكوكيز.

Methods to Prevent Session Hijacking: To be Followed by Web Users

عند استخدام الإنترنت، تأكد من حماية التطبيقات الخاصة بك واختيار المواقع المخولين فقط للتصفح التي تضمن لك حماية البيانات الخاصة بك. بعض التدابير الوقائية الواجب اتباعها أثناء تصفح الإنترنت تشمل:

- لا تنقر على الروابط التي يتم تلقيها من خلال رسائل البريد الإلكتروني أو **LMS**.
- استخدام الجدران النارية لمنع المحتويات الضارة من الدخول إلى الشبكة.
- استخدام إعدادات جدار الحماية والمتصفح لتقييد الكوكيز.
- تأكد من أن الموقع معتمد من قبل سلطات التصديق.
- تأكد من إزالة **history**، **offline content**، وملفات الكوكيز من المتصفح الخاص بك بعد كل معاملة سرية وحساسة.
- تفضيل **HTTPS**، الانتقال الآمن، بدلا من **HTTP** عند إرسال البيانات الحساسة والسرية.
- تسجيل الخروج من المتصفح عن طريق النقر على زر تسجيل الخروج بدلا من إغلاق المتصفح.

IPSec

IPSec هو اختصار لـ **IP security**. وهو يشير إلى مجموعة من البروتوكولات التي تدعم تبادل الحزم الآمنة في طبقة **IP**. وهذه هي تكنولوجيا **VPN** المنتشرة على نطاق واسع لمعالجة التوثيق، والسرية، والنزاهة، والإدارة الرئيسية في شبكات **IP**. **IPsec** يقدم حماية الاتصالات عبر شبكات **IP** بمساعدة تشفير الأجهزة الأمنية.

للحصول على الوظائف المناسبة لـ **IPSec**، يجب على كلا من أجهزة الإرسال والاستقبال تبادل المفتاح العمومي. عادة، يتحقق ذلك من خلال استخدام (**ISAKMP/Oakley**) **Internet Security Association and Key Management Protocol/Oakley**. هذا البروتوكول يسمح لجهاز الاستقبال بالحصول على المفتاح العام ومصادقة المرسل على أساس الشهادات الرقمية "**digital certificates**".

الفوائد التي يقدمها IPSec ما يلي:

- حماية الإعادة "**replay protection**".
- سرية البيانات (التشفير).
- سلامة البيانات.
- مصادقة أصل البيانات.
- مصادقة الند على مستوى الشبكة "**Network-level peer**".

Modes of IPSec

IPSec modes يرتبط مع وظيفة اثنين من البروتوكولات الأساسية، وهما **encapsulating security payload (ESP)** و **Authentication Header (AH)**. كل من هذه البروتوكولات توفر الحماية بإضافة مخطط البيانات إلى الرأس "**header**". الفرق بين الوضعين من التشفير هو أجزاء مخطط بيانات **IP** التي يتم حمايتها وكذلك من حيث ترتيب الرؤوس. **IPsec** يدعم وضعين من التشفير، وهي **transport mode** و **tunnel mode**.

- **وضع النقل "Transport Mode"**

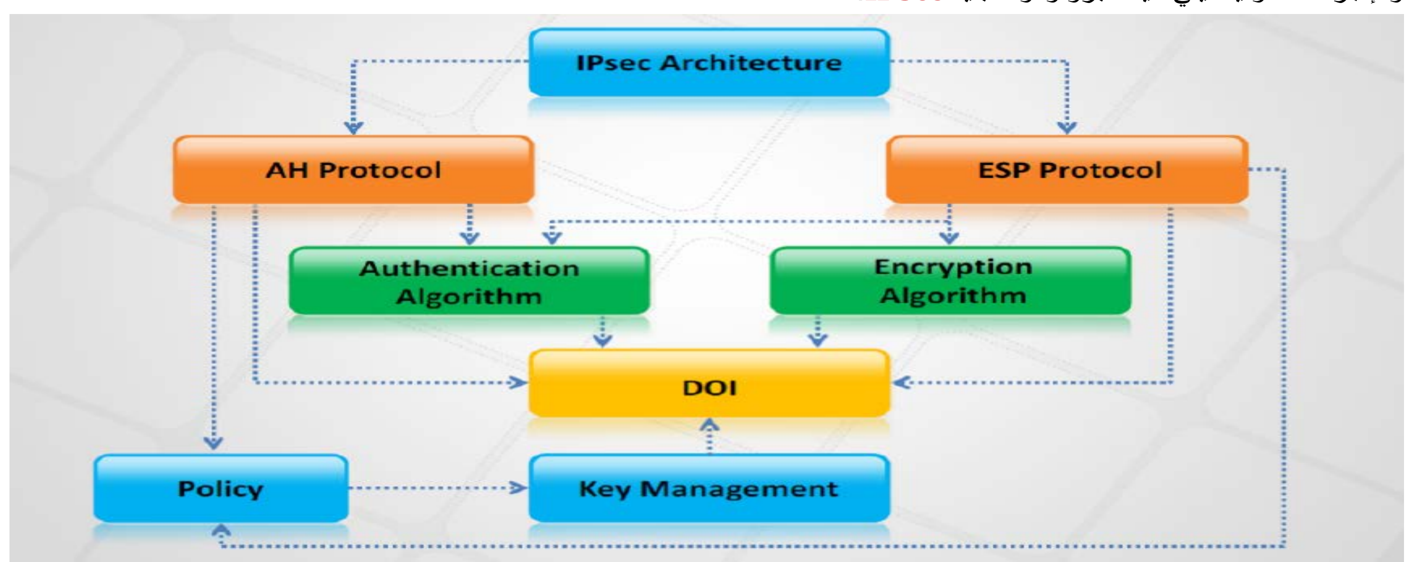
في وضع النقل "**Transport Mode**"، **IPsec** يقوم بتشفير كل حزم الحمولة وترك الرأس لم يمسها. كما انه يسمى ايضا **ESP** (**Encapsulating Security Payload**). فإنه يصادق جهازي كمبيوتر متصلة وله أيضا خيار تشفير نقل البيانات. وهو متوافق مع **NAT**. لذلك، فإنه يمكن استخدامها لتقديم خدمات **VPN** لشبكة باستخدام **NAT**.

- **وضع النفق "Tunnel Mode"**

في وضع النفق "**Tunnel Mode**"، **IPsec** يقوم بتشفير كل من الحمولة والرأس. وبالتالي، يعرف وضع النفق بانه أكثر أمنا. ويسمى وضع النفق أيضا **AH** (**Authentication Header**). سيتم فك تشفير البيانات المشفرة من قبل **IPSec-compliant device** على الجانب المتلقي. **NAT** غير قادر على إعادة كتابة رأس **IP** مشفرة "***encrypted IP header***" وكوضع النفق حيث يتم تشفير رأس حزمة **IP** فأنها ليست قادرة على توفير خدمات **VPN**.

معمارية IPSec

IPSec يقدم خدمات الأمن في طبقة الشبكة. وهذا يعطي حرية اختيار البروتوكولات الأمنية المطلوبة، وتحديد الخوارزميات المستخدمة للخدمات. لتوفير الخدمات المطلوبة توظف مفاتيح التشفير المقابلة إذا لزم الأمر. الخدمات الأمنية التي تقدمها **IPSec** تشمل: التحكم في الوصول، توثيق أصل البيانات، السلامة بدون اتصال، و**antireplay** والسرية. لتحقيق هذه الأهداف، يستخدم **IPSec** بروتوكولين أمن **AH** (**Authentication Header**) و **ESP** (**Encapsulating Security Payload**) وبروتوكولات إدارة المفاتيح المشفرة والإجراءات. وفيما يلي هيكل بروتوكول لبنية **IPSec**:



Encapsulating Security Payload (ESP): تستخدم أساسا لتقديم الخدمات مثل التشفير والتوثيق.

Authentication Header (AH): يستخدم لتوفير خدمة المصادقة لمخطط بيانات الوحيد ولا يوفر التشفير.

DOI: يحدد صيغ الحمولة، **types of exchange**، و **naming conventions** لأمن المعلومات مثل سياسات خوارزمية التشفير أو الأمن. بالإضافة إلى طبقة **IP**، تم تصميم **ISAKMP** لدعم الأجهزة الأمنية في جميع الطبقات. وبالتالي **IPSec** يحتاج إلى **DOI** محدد.

ISAKMP (Internet Security Association and Key Management Protocol): هو عبارة عن بروتوكول المفتاح في بنية **IPSec**. يحدد الأمن المطلوب للاتصالات المختلفة على شبكة الإنترنت مثل الحكومة، القطاع الخاص، والتجاري، وذلك من خلال الجمع بين مفاهيم الأمن من التوثيق، وإدارة المفاتيح والارتباطات الأمنية.

Policy: **Policy** هي العنصر الرئيسي الذي يحدد ما إذا كان كيانين يمكن التواصل مع بعضهم البعض أم لا. إذا كان بإمكانهم التواصل، ثم ما هو نوع التحويل الذي يجب استخدامها؟ إذا لم يتم تعريف السياسة بشكل صحيح، فان الكيانين قد لا يكونا قادرين على التواصل مع بعضها البعض.

IPSec Authentication and Confidentiality

IPSec يستخدم اثنين من الأجهزة الأمنية المختلفة للمصادقة والسرية:

Authentication Header (AH): توفر مصادقة البيانات من المرسل. فهو يستخدم لتوفير السلامة بدون اتصال وتوثيق أصل البيانات لمخططات **IP** وتوفير الحماية ضد **replays**. أنه يوفر مصادقة لرأس **IP** "***authentication for the IP header***" جنبا إلى جنب مع بيانات بروتوكول المستوى التالي. في **IPSec** تتضمن توثيق البيانات مفهومين: سلامة البيانات "**data integrity**" وتوثيق أصل البيانات "**data origin authentication**". توثيق البيانات تشير إما إلى السلامة "**integrity**" وحدها أو إلى كل من هذه المفاهيم، أيضا وتوثيق أصل البيانات "**data origin authentication**" يعتمد على سلامة البيانات "**data integrity**".

- سلامة البيانات "**data integrity**" -تحقق من أن البيانات لم يتم تغييرها.
- توثيق أصل البيانات "**data origin authentication**" -تحقق من ان البيانات أرسلت فعلا من قبل المرسل المطالب بها.

Encapsulation Security Payload (ESP): بالإضافة إلى التوثيق، والسلامة، والحماية ضد أي من هجوم **replay attack**، فان **ESP** يوفر السرية (التشفير). ويمكن استخدامها وحدها أو بالاشتراك مع **AH**. لأنه يحمي فقط حمولة بيانات **IP** بالإعداد الافتراضي. في وضع النفق "**tunnel mode**" يحمي كل من الحمولة ورأس **IP**.

Components of IPSec "عناصر بروتوكول IP الأمني"

يتكون IPSec من العناصر التالية:

IPDec Driver: هذا هو البرنامج الذي يؤدي وظائف على مستوى البروتوكول ومطلوب في التشفير، فك تشفير، المصادقة، والتحقق من الحزمة.

Internet Key Exchange (IKE): **IKE** هو بروتوكول **IPSec** التي تنتج مفاتيح الأمان لـ **IPSec** وغيرها من البروتوكولات.

Internet Security Association Key Management Protocol (ISAKMP): **ISAKMP** هو بروتوكول **IPSec** التي تسمح لاثنين من أجهزة الكمبيوتر التواصل من خلال تشفير البيانات باستخدام الإعدادات الأمنية المشتركة. انه يؤمن أيضا تبادل المفاتيح.

Oakley: **Oakley** هو بروتوكول يستخدم خوارزمية **DIffie-Hellman algorithm** لخلق المفتاح الرئيسي والمفتاح الغير محددة لكل جلسه في نقل البيانات في **IPSec**.

IPSec Policy Agent: هو عبارة عن سلسلة من ويندوز 2000 الذي يجمع إعدادات نهج **IPSec** من **Active Directory** ويحدد الاعداد عند بدء التشغيل.

تنفيذ IPSec "IPSec Implementation"

تنفيذ **IPSec** يتضمن تكرار مختلف مكونات **IPSec**، الواجهات المقدمة من المكونات، معالجة الحزمة الواردة والصادرة. عادة، يختلف تنفيذ **IPSec** التي تعتمد على منصة "**platform**". نحن هنا سوف نناقش تنفيذ **platform-independent IPSec**. معظم تطبيقات **IPSec** تحدد مجموعة من المكونات والتي تشمل:

IPSec base protocols, SADB, SPD, manual keying, ISAKMP/IKE, SA management, and policy management.

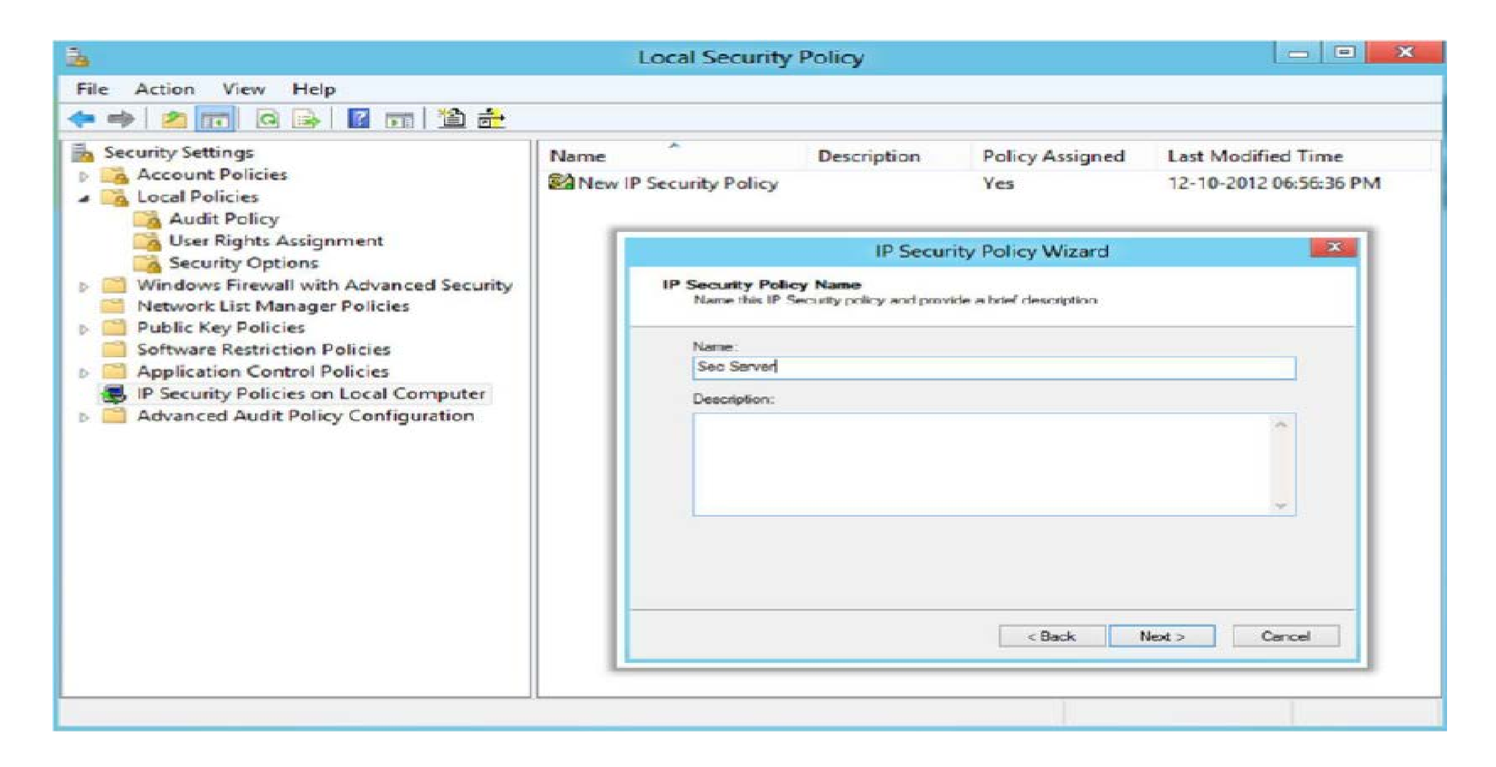
باعتبارك منفذة لـ **IPSec** فيجب عليك أن تكون على علم بجميع هذه المكونات.

IPSec base protocols: يقوم بتنفيذ **ESP** و**AH**. يقوم بمعالجة الرؤوس "**header**"، ويحدد أمن الحزمة من خلال التفاعل مع **SPD** و**SADB**. كما أنه يتعامل مع **fragmentation** و**PMTU**.

SADB: يحافظ على قائمة بـ **SAs** النشطة لكلا من معالجة الواردة والصادرة. و**SADB** يدعم **population of SAs** إما يدويا أو بمساعدة نظام **automatic key management** مثل **IKE**.

SPD: يحدد أمن الحزمة. انه يشير إلى كلا من معالجة الحزم الواردة والصادرة. من أجل التحقق ما إذا كان الأمن الممنوح لحزمة يلبي تكوين الأمان في سياسة **IPSec base protocol component consults the SPD**. وبالمثل في معالجة الصادرة، بروتوكول **IPSec base protocol consults SPD** يقوم بتحديد ما إذا كان الحزمة الصادرة تحتاج أي درجة من الامن.

Internet Key Exchange: عادة ما يعتبر تبادل مفتاح الإنترنت عملية على مستوى المستخدم في جميع أنظمة التشغيل ولكن ليس في أنظمة التشغيل المضمنة. في أجهزة الراوتر (مثال على عقدة في الشبكة) مع أنظمة التشغيل مضمنه، هنا لا يوجد تمييز بين **user space** و**kernel space**. **Policy engine** يقوم باستدعاء **IKE** عندما تكلف السياسة **SA** أو عند وجود حزمة SA ولكن لم يتم تأسيس **SA**.
Peer تقوم أيضا باستدعاء **IKE** عندما تحتاج العقدة التواصل بشكل آمن.
Policy and SA management: تطبق لإدارة السياسات و**SA**.



11.6 اختبار الاختراق (penetration test)

حتى الآن، قد ناقشنا اختطاف الجلسة والمخاطر الناجمة عنها، والتقنيات المختلفة المستخدمة من قبل المهاجمين، والأدوات التي تساعد في اختطاف الجلسة، والتدابير المضادة التي تقدم الحماية ضد اختطاف الجلسة. على افتراض الان أنه أصبح مألوفا مع كل الموضوعات التي تمت مناقشتها سابقا، دعونا نمضي قدما إلى اختبار الاختراق. يسرد هذا القسم ويصف الخطوات المتبعة في اختبار اختراق اختطاف الجلسة.

اختبار اختراق اختطاف الجلسة ينطوي على نفس العملية التي يقوم بها المهاجم من هجوم اختطاف الجلسة. لذا، أولا يجب على مختبر الاختراق على تحديد موقع الجلسة. ثم التحقق من مختلف الاحتمالات لاختطاف الجلسة. ويمكن أن يختلف اعتمادا على الشبكة والآليات التي يستخدمونها في الاتصال. ولكن هنا هو الإجراء القياسي لاختبار الاختراق لاختطاف الجلسة:

- **الخطوة 1: تحديد موقع جلسة**

كما سبق ذكره، فإن الخطوة الأولى هي تحديد جلسة عمل نشطة مستهدفة من خلال عملية التنصت من أجل السيطرة على أكثر من ذلك، وذلك ببساطة لخطف الجلسة.
بعد تحديد الجلسة، تحقق ما إذا كان يتم استخدام معرف جلسة العمل في **URL**. إذا تم استخدام معرف جلسة، تحقق ما إذا كان يتم تشفير الجلسة. إذا لم يتم استخدام معرف جلسة، قم بالتنصت على حركة مرور جلسة العمل بين جهازين.

- **الخطوة 2: التنصت على حركة مرور جلسة العمل بين جهازين**

التنصت على حركة مرور جلسة العمل بين جهازين باستخدام مختلف الأدوات المتاحة مثل الوايرشارك، **CACE pilot**، **Windump**، **Capsa network analyzer**، الخ. شاهد حركة مرور الجلسة والاستيلاء على جلسة حركة مرور الشبكة الضحية. الآن تحقق ما إذا كان يتم تشفير الجلسة أم لا. إذا تم تشفير الجلسة، قم بإفشال الدورة أو استخدام أحصنة طروادة لأداء اختطاف الجلسة. إذا لم يتم تشفير الجلسة، قم **Recover session ID**.

الخطوة 3: استرداد معرف الجلسة "Recover session ID"

إذا لم تكن قادرا على استرداد معرفات الجلسة من جلسة غير مشفرة، فقم باستخدام الأدوات الآلية مثل **Paros proxy**، **Burp Suite**، **Webscarab**، وما إلى ذلك لخطف الجلسة. باستخدام هذه الأدوات يجعل عملية اختطاف الجلسة سهلة.
إذا تم استرجاع معرف الجلسة "**session ID**"، يتم التحقق ما إذا كان يتم تشفيرها أم لا. عادة يتم إنشاء معرفات جلسة العمل باستخدام خوارزميات مختلفة. إذا كان المهاجم قادرا على التنبؤ بالخوارزميات المستخدمة لتوليد معرف الجلسة، فإنه يمكنه بسهوله تحديد أو استرداد معرفات الجلسة. إذا تم تشفير معرف الجلسة، قم بكسر تشفير معرف الجلسة وإذا لم يتم كسر تشفير معرف الجلسة، فيمكنك إرسال رسائل التصيد للضحية من أجل أداء **session fixation**.

الخطوة 4: كسر تشفير معرف جلسة

يتم كسر معرف الجلسة إذا تم ترميز **URL**، **HTML encoded**، **unicode encoded**، **Base64 encoded**، أو **hex encoded**. تكسير معرفات الجلسة المشفرة يعطي معرفات جلسة العمل الأصلية للضحية. عادة ما تكون معرفات الجلسات هي المسؤولة عن مصادقة المستخدم. إذا كنت قادرا على استرداد معرفات جلسة المستخدم الأصيلة، حينها يمكنك حقن نفسك في الجهاز بين الضحية والجهاز البعيد، ويمكن استخدام هذا الصدد غير المصرح به لأغراض خبيثة خاصة بك.
بمجرد النجاح في كسر **session ID encryption**، فانه يمكنك تنفيذ **session fixation** مع مساعدة من **phishing mails**.

الخطوة 5: إجراء اتصال عادي مع واحدة من آلات

بعد تنفيذ **session fixation** يمكنك إجراء اتصال عادي مع واحدة من آلات التي وجدت في حركة مرور الشبكة ويمكن الوصول إلى الأجهزة البعيدة عن طريق اخفاء نفسك كمستخدم مصرح به للشبكة.

الخطوة 6: جمع معرفات الجلسات العديدة

بمجرد إنك قمت بالاتصال بواحد من الآلات في الشبكة، يمكنك جمع معرفات الجلسة العديدة. هناك نوعان من التقنيات المختلفة المتاحة لاسترجاع معرفات الجلسة. وهم من خلال الكوكيز في **response headers**، وعن طريق مطابقة **regular expression** مقابل **response body**. لجمع معرفات الجلسة من ملفات الكوكيز، يجب عليك التأكد من ألا يتم تحديد خانة الاختيار "**from message body**" وعندما يتم جمع معرفات الجلسات من نص الرسالة، يجب عليك التأكد من أن يتم تحديد خانة الاختيار "**from message body**".

الخطوة 7: التنبؤ بمعرف جلسة العمل الجديدة

الآن قم بتحليل معرفات الجلسات التي تم جمعها من التنبؤ أو تخمين معرف جلسة العمل الجديدة. يجب عليك التنبؤ بمعرف جلسة العمل الجديدة من أجل العثور على معرف جلسة الحالي وأداء هجوم **replay attack**.

الخطوة 8: إعادة معرف الجلسة الجديدة "Replay new session ID"

هجوم **replay attack** يحدث عند نسخ سيل من الرسائل (معرفات الجلسات) بين طرفين وإعادة التيار إلى واحد أو أكثر من الأطراف. ما لم يتم التخفيف، فان أجهزة الكمبيوتر تخضع لعملية الهجوم، مما يؤدى إلى مجموعة من العواقب السيئة.
الآن تحقق لإنشاء اتصال. إذا تم تأسيس الاتصال، فعليك توثيق جميع نتائج اختبار الاختراق. إذا لم يتم تأسيس الاتصال، قم بتطبيق تقنية **brute forcing** من أجل إيجاد معرف الجلسة الصالحة الحالي لتأسيس الاتصال.

الخطوة 9: Brute force session IDs

قم بأداء **Brute force session IDs** مع نطاق ممكن من قيم معرف الجلسة المحدود، حتى يتم العثور على معرف الجلسة الصحيح. وهذا ينطوي على جعل الآلاف من الطلبات باستخدام كل معرفات الجلسة التي تم إنشاؤها بشكل عشوائي. هذه التقنية شاملة ولكن تأخذ وقتا طويلا.

الخطوة 10: توثيق جميع النتائج

الخطوة الأخيرة في أي اختبار اختراق وهو توثيق جميع النتائج التي تم الحصول عليها من خلال كل الاختبارات.

الحمد لله تعالى، وبحول الله تعالى نكون قد انتهينا من الوحدة الحادية عشر والتي سوف نتطرق لما ذكر في هذه الوحدة كثيرا وذلك في الوحدات القادمة من CEHv8.ونلقاكم مع الوحدة التالية:

د. محمد صبحي طيبه